Quick Start ArcGIS Enterprise with Automation

Shannon Kalisky
Mark Carlson
Nikhil Shampur
Cherry Lin
“Software installation is my favorite! I really love how complex and repetitive it can be - especially when my entire organization is depending me to get it done quickly!”

- No one ever
Deployment Automation
How you setup the software

Workflow Automation
How you automate analytical and administrative tasks

Automation Tools

- ArcGIS Enterprise Builder
- Amazon Web Services
- Microsoft Azure
- Chef
- Powershell DSC
- ArcGIS API for Python
- webgisdr Utility
- and more
Deployment automation tools styles/experiences

Wizard:
- ArcGIS Enterprise Builder

Machine Images + Tooling:
- Amazon Web Services
- Microsoft Azure

Script-based:
- Chef
- Powershell DSC
Best fit environments for deployment automation tools

Cloud deployments

- AWS
- Microsoft Azure
- Cloud deployments
- Amazon Web Services
- Chef
- PowerShell
- DSC
- ArcGIS Enterprise Builder
Best fit environments for deployment automation tools

On-premises deployments

All-in-one, single machine deployments **only**

ArcGIS Enterprise Builder

Amazon Web Services

Microsoft Azure

Any deployment scenario

Chef

Powershell DSC
Tools that have deep dive sessions

Tools that have a deep dive session we will overview quickly

The Builder deep dive session has already passed - please bring questions to the Expo.
Why use one tool over another?
**ArcGIS Enterprise Builder | Prerequisites**

- Review documented system requirements

- Obtain license files for ArcGIS Server and Portal for ArcGIS

- Ensure HTTPS is enabled on your web server
ArcGIS Enterprise Builder | Architecture

- Single Machine Deployment
Welcome to the ArcGIS Enterprise Builder

This builder will install a base deployment on a single machine.

Master Agreement

You must agree with the master agreement below to proceed.

I accept the master agreement
I do not accept the master agreement

Ready To Install

Your ArcGIS Enterprise Installation is ready to begin.
Welcome!

You are close to completing the configuration of your ArcGIS Enterprise base deployment.

To complete the configuration, the browser must automatically switch to a secure (HTTPS) connection and display a warning message that you will need to advance through.

Click Next to continue.

Initial Administrator Account

Create the account that will be used as the initial administrator for your ArcGIS Enterprise deployment.

This is a new account that is stored with your deployment and is not an operating system account. Use this account when you are asked to sign in to your deployment for the first time and for administrative tasks such as creating additional user accounts.

Username: admin
Password: *********
Confirm Password: *********

Additional Account Information

Enter additional information for the initial administrator account you specified on the previous prior panel.

- First Name: admin
- Last Name: lastname
- Email: myemail@domain.com
- Confirm Email: myemail@domain.com
- Security Question: What city were you born in?
- Security Answer: Anytown

Root Content Directory

Your ArcGIS Enterprise deployment requires a content directory for storing data and configuration files. Your deployment is going to grow, so you should specify a directory location with at least 10 GB of free space to start with.

Content Directory: C:\arcgis

Configuration Summary

Username: admin
Content Directory: C:\arcgis

Click Finish to complete the configuration of your deployment.
ArcGIS Enterprise Builder | Current 10.6 Limitations

- Does not support multi-machine deployments

- Unable to define your own name for the web adaptors

https://enterprise.arcgis.com/en/get-started/latest/windows/arcgis-enterprise-builder.htm#ESRI_SECTION1_3373BE50D5134A6AACF84A03531B613A
ArcGIS Enterprise on AWS

Automation
### Getting Started

<table>
<thead>
<tr>
<th><strong>On-premise</strong></th>
<th><strong>AWS</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Esri authorization files</td>
<td>Esri authorization files</td>
</tr>
<tr>
<td>Domain name for your application</td>
<td>Domain name for your application</td>
</tr>
<tr>
<td>SSL Certificate for your domain</td>
<td>SSL Certificate for your domain</td>
</tr>
<tr>
<td>Esri software setups</td>
<td>AWS Account</td>
</tr>
<tr>
<td><strong>Infrastructure</strong></td>
<td></td>
</tr>
<tr>
<td>Machines/VMs/Networks</td>
<td></td>
</tr>
<tr>
<td>Web Servers and/or Load Balancer</td>
<td></td>
</tr>
<tr>
<td>File Server</td>
<td></td>
</tr>
<tr>
<td>Data storage/Database</td>
<td></td>
</tr>
<tr>
<td>......</td>
<td></td>
</tr>
</tbody>
</table>
AMIs

- Windows with SQL Server Express
- Ubuntu with Postgresql
Esri Cloud Formation Templates

CloudFormation templates to deploy ArcGIS Enterprise on Amazon Web Services

The following templates use CloudFormation to create an ArcGIS Enterprise deployment or ArcGIS Server roles on Amazon Web Services (AWS).

1. Before using a template, read the README file for the sample template you want to use. Make sure you follow the directions to set up the required components first.
2. Click LAUNCH STACK to use a template. You will be taken to the AWS Console. Sign in to your Amazon Web Services account, and enter the required parameters for the template.
3. For more information, see the ArcGIS Enterprise on Amazon Web Services and ArcGIS Enterprise on Windows or ArcGIS Enterprise on Linux.

ArcGIS Enterprise needs to run in a VPC environment. Most Amazon Web Services accounts have a default AWS VPC created. You use this default VPC, create another VPC manually, or use the template described in the next section to create a VPC. However, you create the VPC it needs to have the following properties set:

- **DNS resolution**: Set to yes.
- **DNS Hostname**: Set to your desired name.
- **DHCP Options set**: You must have both domain-name and domain-name-servers set, for example: domain-name=esri.int, domain-name-servers=esri-internet-provided

Template for Deploying AWS Roles

Portals for ArcGIS

Hosting Server

ArcGIS Web Adaptor

ArcGIS Data Store (relational + tile cache)

GeoAnalytics Server

GIS Server

Image Server

GeoEvent Server
Native to AWS Tools

Automate CloudFormation Stack creation using Python

Cloudformation stack creation can be run from command line. You can check Amazon Cloud

Below is a sample python script to create cloudformation stacks. To run it:
- Make sure you have Python installed. Python is also installed with ArcGIS Desktop and ArcGIS Server.
- Download and install AWS SDK for Python (boto). The easiest way is to install via Pip.
- If you don’t have Pip, install Pip following the directions at https://pip.pypa.io/en/stable/installing.html.
- Install Boto by running: pip install boto
- Download the sample python script and run it with the parameters required by your cloudformation templates.
- For example, ArcGIS for Server WebGIS parameter file of parameters.xml defines the parameters required to create a cloudformation stack creation.py
- Your AWS Access key (Your AWS Secret Access Key

Automate CloudFormation Stack creation using Powershell.

CloudFormation stack creation can also be automated using Powershell. Two sample powershell scripts are provided:
- arcgis-webgis-ha-windows.ps1 to create highly available WebGIS stack on Windows.
- arcgis-server-ubuntu-ha-configstore.ps1 to create Server GIS site using DynamicOS and SS as config-store.

Esri-provided sample scripts

Android  iOS  Java  JavaScript  .NET  Node.js  PHP

Deployment Tool - Cloud Builder Command Line Interface for AWS

- New at 10.6

- One Run to Deploy ArcGIS Enterprise
  - Base + Federated Servers
  - `cloudbuilder.exe CREATE -j <properties.json file>

- Prepare your deployment
Cloud Formation Templates VS. Cloud Builder CLI

- Highly Customizable
- One Run Deployment
- Power
- Simplicity
Deep Dive Session

ArcGIS Enterprise: Cloud Operations Using Amazon Web Services

Date: 03/08/2018

Time: 4:00pm - 5:00pm

Location: Mojave Learning Center
ArcGIS Enterprise on Azure

Automation
Cloud Builder Overview

- Wizard driven deployment experience
- Base deployment + Distributed GIS
  - Single Machine/Tier
  - Multi Machine/Tiers

- New at 10.6
  - Designer to generate automation artifacts
Deployment Options

- Cloud Builder
- Visual Studio
- PowerShell
- Command Line

- Windows
- Mac
- Linux
Integrate with …

Azure Cloud Shell

PowerShell DSC

CHEF
puppet labs
SALTSTACK
ANSIBLE
Octopus Deploy
VAGRANT
TeamCity
Travis CI
Cloud Builder VS. Automation

- Visual Studio
- BASH
- PowerShell

Customize or Extend
Wizard Driven

Power
Simplicity
Deep Dive Session

ArcGIS Enterprise: Cloud Operations Using Microsoft Azure

Date: 03/08/2018

Time: 5:30pm - 6:30pm

Location: Mojave Learning Center
Automation with Chef
Esri and Chef

- ArcGIS Cookbook first released at 10.3.1

- Chef
  - One of the most popular IT standard automation framework
  - Install ready-to-use cookbooks and define the configuration
  - Run chef with the configuration file
  - Deploy exact way you want
  - No programming skills required
Use Esri Chef Automation When ……

- **Automation**
  - Reliable and testable deployments and upgrades
  - Faster Disaster Recovery
  - Moving between data centers
  - Rapid Auto-Scaling

- **Chef**
  - Ready-to-use tools with minimum requirements to start
  - Across environments and platforms
Easy to Start

Install Chef Client and Download Cookbook

Edit Property json file

"chef-solo -j <properties>.json"
Upgrade

Download New Version of Cookbook

New Version #; New Setups; New License Files

"chef-solo -j <properties>.json"
Demo
Install and Upgrade ArcGIS Enterprise Using Chef
Getting Started

http://esri.github.io/arcgis-cookbook/
Welcome to Chef on ArcGIS Enterprise!
Here, you’ll find all you need to automate your Web GIS installation and administration.

ArcGIS Enterprise 10.6 is available now!
See what’s new in 10.6

Get Started

What is ArcGIS Enterprise?
ArcGIS Enterprise delivers industry-leading mapping and data analysis to your physical or cloud infrastructure.
The software platform can be deployed in a variety of configurations – whatever best suits your organization.

Get acquainted with Enterprise

What is Chef?
Chef is a software framework to automate installation and configuration of IT components.
It uses a Chef server, which can be deployed open-source, as SaaS, or with a subscription to Chef Automate.

Learn more about Chef
Highly Configurable and Flexible

Single-machine deployment

Multi-machine deployment

ArcGIS GeoAnalytics Server

ArcGIS Image Server

Highly available deployment
Varieties of Environments
ArcGIS Software

- ArcGIS Enterprise
  - Portal for ArcGIS
  - ArcGIS Server
    - Enabled Roles: GIS Server, Image Server, GeoAnalytics Server, GeoEvent Server
  - ArcGIS Data Store
    - relational, tile cache, spatiotemporal big data store
  - ArcGIS Web Adaptor
- And other Esri Software:
  - ArcGIS Pro
  - Insights for ArcGIS
  - ArcGIS Desktop
  - ArcGIS License Manager
Best Practices
Plan Ahead

- Need plan ahead
  - Design
  - Configuration
  - Testing
- Repeatable
  - Success in production environment
  - Less downtime in production
  - Faster recovery
Consider Baking a Base Image

- Install Chef; Cookbook
- Have Some Processes Done. e.g., Setups, System Requirements
- Have some components baked in, e.g., base map, security updates, etc
Work With

- GIS workflow automation tools
  - Python API for ArcGIS
  - WebGISDR tool

- Whole system infrastructure automation
  - High Availability and Disaster Recovery deployment
  - Moving to different data centers

- Other software deployment automation tools
Other Tricks & Tips

• Upgrade:
  • Use Chef to deploy first

• Disconnected Environment Considerations
  • Make sure you have proper Esri license files staged for use
  • Software setups in a common location for access
  • Having Chef Client installation staged for installation
Other Tricks & Tips

- Set password in an environment variable
- "-l debug" for debug
- Attributes value of True or False, no "". e.g. "configure_autostart": true
- ArcGIS Cookbooks support enabling Server roles.
- Support security configuration for ArcGIS Server.
Key Points about ArcGIS Chef Automation

- Doesn’t require programming skills
- Easy to start
- Flexible and highly configurable
- Repeatable
- Plan ahead
Automation with PowerShell DSC
Use PowerShell DSC when …

- Microsoft Shop
  - Bias towards Microsoft Windows Tools
  - PowerShell DSC is built into Windows
- Windows Administrators love PowerShell
  - DSC is a natural fit
- Low Tech solution for push automation across multiple machines
  - No centralized server needed
PowerShell over the years

PowerShell Releases

1.0 Nov 2006
2.0 Oct 2009
3.0 Nov 2012
4.0 Oct 2013
5.0 Apr 2014
6.0 Jan 2018

Windows 7
Windows Server 2008
Windows Server 2008 R2
Windows Server 2012
Windows Server 2012 R2
Windows 8
Windows 8.1
Windows 10
Windows Server 2016
Linux
What is PowerShell DSC (Desired State Configuration)

“Declarative platform used for configuration, deployment, and management of systems.”

Automation Concepts
- Idempotent
- Declarative

Not Idempotent

```
New-SmbShare -Path 'foo\share'
```

Imperative

```
Import-Module ServerManager
#Check and install ASP.NET 4.5 feature
if (-not (Get-WindowsFeature "Web-Asp.Net45").Installed) {
    try {
        Add-WindowsFeature Web-Asp.Net45 -ErrorAction Stop
    } catch {
        Write-Error $_
    }
}
#Check and install Web Server Feature
if (-not (Get-WindowsFeature "Web-Server").Installed) {
    try {
        Add-WindowsFeature Web-Server -ErrorAction Stop
    } catch {
        Write-Error $_
    }
}
#Create a new website
Import-Module WebAdministration
New-WebSite -Name MyWebSite -Port 80 -HostHeader MyWebSite -PhysicalPath "$env:systemdrive\inetpub\wwwroot\MyWebSite"
Start the website
Start-WebSite -Name MyWebSite
```

Add Checks to ensure idempotent

```
if(-not((Get-SmbShare -Name 'foo\share'))) {
    New-SmbShare -Path 'foo\share'
}
```

Declarative
PowerShell DSC

```powershell
Configuration IISWebsite
{
    Node Server1, Server2
    {
        WindowsFeature IIS
        {
            Ensure = "Present"
            Name   = "Web-Server"
        }
        WindowsFeature ASP
        {
            Ensure = "Present"
            Name   = "Web-Asp-Net45"
        }
    }
}
```

### Built in DSC Resources

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive</td>
<td>Unpack .zip files</td>
</tr>
<tr>
<td>Environment</td>
<td>Manage env variables</td>
</tr>
<tr>
<td>File</td>
<td>Manage files, folders</td>
</tr>
<tr>
<td>Group</td>
<td>Manage local groups</td>
</tr>
<tr>
<td>Log</td>
<td>Write message to log</td>
</tr>
<tr>
<td>Package</td>
<td>Install/Uninstall .msi and setups</td>
</tr>
<tr>
<td>Registry</td>
<td>Manage registry keys &amp; values</td>
</tr>
<tr>
<td>Script</td>
<td>Run PowerShell script blocks</td>
</tr>
<tr>
<td>Service</td>
<td>Manage Services</td>
</tr>
<tr>
<td>User</td>
<td>Manage local users</td>
</tr>
<tr>
<td>Windows Feature</td>
<td>Add/Remove Windows Features</td>
</tr>
<tr>
<td>Windows Process</td>
<td>Manager Processes</td>
</tr>
</tbody>
</table>

**Start-DscConfiguration 'server' -Verbose -Wait**
Wealth of DSC Resources

xHyper-V
xSafeHarbor
xSmbShare
xDismFeature
xFailoverCluster
xSQLServer
xComputerManagement
x7Zip
xRobocopy
xWindowsUpdate
xWindowsConainer
xWebDeploy
xWordPress
xDBase
xMySQL
xFirewall
xDisk
xChrome
xDefender
xPython
xPHP
xWinEventLog
xWindowsRestore
xSharePoint
xBitLocker
What about ArcGIS?

Wouldn’t this be great?

Not so Fast!
GIS Infrastructure needs vary.
Discrete States

- Install Software
- License Software
- Configure Service Account
  - Create Windows Local/Domain User Account
  - Assign File System Privileges
  - Assign Windows Service to User
- Create Site
- Import SSL Certificates
- Register/Configure Databases or Data Stores
- Federate Site
- Stop/Start Server Services
  - Publish Services
## ArcGIS DSC Resources

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArcGIS_Install</td>
<td>Manages the (un)installation of ArcGIS (Server) Software</td>
</tr>
<tr>
<td>ArcGIS_License</td>
<td>Licenses ArcGIS (Server) Software</td>
</tr>
<tr>
<td>ArcGIS_ServiceAccount</td>
<td>Configures the ‘Run As’ Service with appropriate (file system) privileges</td>
</tr>
<tr>
<td>ArcGIS_Server</td>
<td>Configures the ArcGIS Server Site</td>
</tr>
<tr>
<td>ArcGIS_Portal</td>
<td>Configures the Portal for ArcGIS</td>
</tr>
<tr>
<td>ArcGIS_DataStore</td>
<td>Configures the ArcGIS DataStore</td>
</tr>
<tr>
<td>ArcGIS_Server_TLS</td>
<td>Configures SSL on a Server Site</td>
</tr>
<tr>
<td>ArcGIS_Portal_TLS</td>
<td>Configures SSL on a Portal Site</td>
</tr>
<tr>
<td>ArcGIS_WebAdaptor</td>
<td>Configures Web Adaptor on IIS for Portal/Server</td>
</tr>
<tr>
<td>ArcGIS_EGDB</td>
<td>Enables/Registers an SQL Server flavor EGDB with Server</td>
</tr>
<tr>
<td>ArcGIS_GeoEvent</td>
<td>Configures GeoEvent Extension for Server</td>
</tr>
<tr>
<td>ArcGIS_Federation</td>
<td>Configures Federation between Server and Portal</td>
</tr>
<tr>
<td>ArcGIS_IIS_TLS</td>
<td>Configures SSL on IIS</td>
</tr>
</tbody>
</table>
Do I need to learn all these DSC Resources to use it?

No!

*We want the getting started experience to be simple*

You only get one first impression
Getting Started

3 Easy Steps
Getting Started

- Install ArcGIS PowerShell Module

```powershell
PS C:\> Install-Module arcgis
```

- Edit (JSON) configuration file
  - Sample Configurations are documented for common site topologies

```json
"configData": {
  "version": "10.5.1",
  "server": "arcgis",
  "portalContext": "portal",
  "serverName": "specialPurposeServer",
  "credentials": {
    "primarySiteAdmin": {
      "username": "admin",
      "email": "admin@domain.com",
      "password": "changeit"
    },
    "serviceAccount": {
      "username": "ArcGISAdmin",
      "password": "ArcGis123456",
      "email": "arcgis@domain.com"
    }
  },
  "server": {
    "licenseFilePath": "\\\local\ArcGIS_Automation\Authorization_Files\\Version10.5\\Server_Ent_Adv_ALLExt.lic",
    "installDir": "C:\\ArcGIS\\\server",
    "installPython": "C:\\Python\python27\python.exe",
    "serverDirectoriesAndLocation": "C:\\ArcGISServer\directories",
    "configLocation": "C:\\ArcGISServer\\config\",
    "portal": {
      "licenseFilePath": "\\\local\ArcGIS_Automation\Authorization_Files\\Version10.5\\portal_2000_1000.lic"
    }
  }
}
```

- Start Configuration

```powershell
PS C:\> Configure-ArcGIS pathToConfigFile
```
Automation using PowerShell DSC | System Requirements

- PowerShell 4.0 or Higher
  - Windows Management Framework 4.0 or Higher

Windows PowerShell 4.0 runs on the following versions of Windows:

- Windows 8.1, installed by default
- Windows Server 2012 R2, installed by default
- Windows® 7 with Service Pack 1, install Windows Management Framework 4.0 to run Windows PowerShell 4.0
- Windows Server® 2008 R2 with Service Pack 1, install Windows Management Framework 4.0

- Automation delivered via
  - PowerShell Gallery (best experience)
  - GitHub (Manual Install)

- Download Installers and Licenses from my ESRI (regular means)
Demo
PowerShell DSC
Automation using PowerShell DSC | Key Takeaways

- New Automation Option at 10.6 (Beta)
  - Analogous to Chef Automation
  - Targets an audience with a bias towards Microsoft Windows tooling
    - Windows Admins love PowerShell
    - DSC - Natural Fit
  - Easy Low Tech Solution for Push Model

- Delivered via
  - PowerShell Gallery https://www.powershellgallery.com/packages/ArcGIS/
  - Github https://github.com/Esri/arcgis-powershell-dsc

- Supports Single and Multi machine sites
  - Install, Uninstall and Upgrade workflows
Summary

Recap of when to use each tool
Best fit use case:

- **ArcGIS Enterprise Builder = On-premises, all-in-one, single machine deployment**
  - Base deployment only

- **AWS = Deploying in Amazon cloud**
  - Base deployment, server roles

- **Azure = Deploying in Azure cloud**
  - Base deployment, server roles

- **Chef = Multi-machine deployments, on-premises or cloud, Windows or Linux, repeat deployments**
  - Base deployment, server roles, HA, DR

- **Powershell DSC = Multi-machine deployments, on-premises - Windows**
  - Base deployment, server roles, HA