

Deployit EC2 Plugin Manual

Version 3.9.2

Table of Contents

Table of Contents	2
Preface	3
Overview	3
Features	3
Requirements	3
Usage scenarios	3
Creating a single host	3
Creating an environment	3
Provisioning instantiated hosts	3
Configuration instructions	4
EC2 credentials	4
Host template	4
Environment template	4
Using the EC2 plugin	4
CI Reference	5
Configuration Item Overview	5
Containers	5
Other Configuration Items	5
Configuration Item Details	5
cloud.BaseHostTemplate	5
cloud.CifsHost	5
cloud.CloudEnvironmentParameters	7
cloud.Environment	7
cloud.EnvironmentTemplate	8
cloud.HostParameters	8
cloud.SshHost	8
ec2.Credentials	10
ec2.HostTemplate	10

Preface

This document describes the functionality provided by the EC2 plugin.

See the **Deployit Reference Manual** for background information on Deployit and deployment concepts.

Overview

The EC2 plugin is a Deployit plugin that supports launching, provisioning and terminating hosts and environments on the EC2 platform.

The EC2 plugin is part of the Deployit **Cloud Pack**. For more information about the Cloud Pack, see the **Deployit Cloud Pack Manual**.

Features

- Launching and terminating EC2 instances from Deployit;
- Combining EC2 instances into environments which can be used for application deployment;
- Registering middleware on EC2 instances as part of a created environment.

Requirements

- **Deployit requirements**
 - **Deployit:** version 3.9+
 - **Other Deployit Plugins:** cloud-plugin
- **Infrastructural requirements**
 - **EC2 credentials:** AWS key and secret for accessing the EC2 platform via its API.

Usage scenarios

This section describes the most common usage scenarios, further sections contain more detailed configuration instructions.

Creating a single host

In its simplest form, the EC2 plugin can launch a single instance on EC2 and register it in Deployit as a CI of type `cloud.SshHost` (when connecting to the host using SSH) or `cloud.CifsHost` (when connecting using CIFS). The resulting host CI can contain middleware CIs that are present on the host and can be used as normal containers for deployment. The host can also be destroyed, which causes Deployit to terminate the EC2 instance and remove the host CI and its children from the repository.

There is a special CI type `ec2.HostTemplate` which is used as a template to define all information about the future host.

Creating an environment

For more information about combining cloud hosts into environments, see the **Deployit Cloud Pack Manual**.

Provisioning instantiated hosts

When launching an instance on EC2, the AMI it is based on may already have the desired middleware installed. If this is the case, a launched host will be ready for use as soon as it has finished booting. It is also possible to provision a host using Puppet, Chef or a shell command after launching it. This is supported via the notion of *marker file*. If the host template specifies a marker file, Deployit will poll the launched instance for its presence. When the file is found on the instance filesystem, Deployit will conclude the host is up and ready for deployment. The location of the marker file can be configured in the `ec2.HostTemplate`.

See the section on the marker file in the **Deployit Cloud Pack Manual** for additional details on the polling process.

Note:

- it is the responsibility of the AMI to invoke the provisioning process and to create the marker file when provisioning is completed, signaling that the host and middleware is ready for deployment.
- marker files are only supported on Unix-based AMIs.

Configuration instructions

EC2 credentials

The EC2 plugin requires access to the AWS key and secret to access the EC2 API. These credentials are specified under the *Configuration* root node in the repository using a `ec2.Credentials` CI. The CI has a control task `validateCredentials` that can test that the credentials can be used to communicate with EC2.

Host template

The next step is to define host template CIs (`ec2.HostTemplate`). A host template describes a single host that can be launched on EC2. In addition to the generic host template properties, it allows some EC2-specific properties.

See **CI reference** for list of `ec2.HostTemplate` properties and their meaning.

See the Cloud Pack Manual for the generic properties.

Here is an example of an EC2 host descriptor:

```
<#escape x as x?xml>
<list>
  <cloud.SshHost id="{hostsPath}/{hostTemplate.name}_{hostAddress}">
    <template ref="{hostTemplate.id}"/>
    <cloudId>{cloudId}</cloudId>
    <address>{hostAddress}</address>
    <#if hostTemplate.privateKeyFile?><privateKeyFile>{hostTemplate.privateKeyFile}</privateKeyFile></#if>
    <#if hostTemplate.username?><username>{hostTemplate.username}</username></#if>
    <#if hostTemplate.password?><password>{hostTemplate.password}</password></#if>
    <#if hostTemplate.os?><os>{hostTemplate.os}</os></#if>
    <#if hostTemplate.connectionType?><os>{hostTemplate.connectionType}</connectionType></#if>
  </cloud.SshHost>
  <www.ApacheHttpdServer id="{hostsPath}/{hostTemplate.name}_{hostAddress}/httpd">
    <host ref="{hostsPath}/{hostTemplate.name}_{hostAddress}"/>
    <startCommand>sudo apachectl stop</startCommand>
    <startWaitTime>3</startWaitTime>
    <stopCommand>sudo apachectl stop</stopCommand>
    <stopWaitTime>3</stopWaitTime>
    <restartCommand>sudo apachectl restart</restartCommand>
    <restartWaitTime>10</restartWaitTime>
    <defaultDocumentRoot>/var/www</defaultDocumentRoot>
    <configurationFragmentDirectory>/etc/apache2/conf.d</configurationFragmentDirectory>
  </www.ApacheHttpdServer>
</list>
</#escape>
```

Every `ec2.HostTemplate` CI provides a `validateDescriptor` control task which processes the Freemarker template, parses the resulting XML and reports errors if something is wrong. No actual changes are made to the repository during execution of this control task.

Please note that:

- EC2 hosts which you define here should be either `cloud.SshHost` or `cloud.CifsHost`.
- The `cloud.SshHost` or `cloud.CifsHost` in the template must contain the XML fragments for `address`, `cloudId` and `template`. These are needed for the proper functioning of the plugin.
- Since XML is being generated you have to make sure that values are properly encoded. You can achieve this by enclosing the template in `<#escape x as x?xml>...</#escape>`, or alternatively use `${exampleKey?xml}`. See the Freemarker documentation for details.

Environment template

For more information about defining environment templates, see the **Deployit Cloud Pack Manual**.

Using the EC2 plugin

Please see the **Cloud Pack manual** for instructions on how to use the environment and host templates provided in the EC2 plugin.

CI Reference

Configuration Item Overview

Containers

CI	Description
cloud.CifsHost	Cloud host with CIFS access
cloud.SshHost	Cloud host with SSH access

Other Configuration Items

CI	Description
cloud.BaseHostTemplate	Base class for instance templates, all instance templates must extend it
cloud.CifsHost	Cloud host with CIFS access
cloud.CloudEnvironmentParameters	Parameters for cloud environment instantiation
cloud.Environment	Cloud environment
cloud.EnvironmentTemplate	Cloud environment template
cloud.HostParameters	Parameters for host templates instantiation
cloud.SshHost	Cloud host with SSH access
ec2.Credentials	Amazon EC2 credentials
ec2.HostTemplate	Amazon EC2 instance template

Configuration Item Details

cloud.BaseHostTemplate

Virtual Type

Interfaces `udm.ConfigurationItem`

Base class for instance templates, all instance templates must extend it

Public Properties			
* bootTimeout :	INTEGER	= 500	Maximal amount of time (in seconds) allowed to elapse before the instance is ready.
* xmlDescriptor :	STRING		Freemarker template of XML which describes instance and middleware
connectionType :	ENUM	[SFTP, SFTP_CYGWIN, SFTP_WINSSHD, SCP, SUDO, INTERACTIVE_SUDO, TUNNEL, TELNET, WINRM, WINRM_HTTP, WINRM_HTTPS]	Connection type to be used for connecting to the host
markerPath :	STRING		Path to the file which should appear on the instance when provisioning completes.
os :	ENUM	[WINDOWS, UNIX]	OS family
password :	STRING		Password
privateKeyFile :	STRING		Private key file to use for authentication
retryDelay :	INTEGER	= 5	Delay (in seconds) after each connection attempt.
username :	STRING		Username
Control task	Parameter CI	Attributes	Description
instantiate	cloud.HostParameters		Create instance from template
validateDescriptor			Validate XML descriptor

cloud.CifsHost

Type Hierarchy	overthere.CifsHost >> overthere.RemoteHost >> overthere.Host >> udm.BaseContainer >> udm.BaseConfigurationItem
Interfaces	udm.Taggable, udm.ConfigurationItem, udm.Container, overthere.HostContainer

Cloud host with CIFS access

Public Properties	
* address : STRING	Address of the host
* cloudId : STRING	Unique ID within cloud platform
* connectionType : ENUM [TELNET, WINRM, WINRM_HTTP, WINRM_HTTPS] = WINRM	Connection Type
* os : ENUM [WINDOWS, UNIX] = WINDOWS	Operating system the host runs
* password : STRING	Password to use for authentication
* template : CI<cloud.BaseHostTemplate >	Template which was used to create this host
* username : STRING	Username to connect with
cifsPort : INTEGER = 445	Port on which the CIFS server runs
jumpstation : CI<overthere.Jumpstation>	Jumpstation that should be used to reach this host
pathShareMappings : MAP_STRING_STRING	Mapping from Windows paths to Windows share names, e.g. C:\IBM\WebSphere -> WebSphereShare
port : INTEGER	Port on which the Telnet or WinRM server runs
tags : SET_OF_STRING	If set, only deployables with the same tag will be automatically mapped to this container.
temporaryDirectoryPath : STRING	Directory into which temporary files are stored. Will be cleaned up when the connection is closed.
winrmEnableHttps : BOOLEAN = false	Enable SSL communication to the WinRM server

Hidden Properties				
* connectionTimeoutMillis	: INTEGER = 1200000		Number of milliseconds Overthere waits for a connection to a remote host to be established	
* protocol	: STRING = cifs		Protocol to use when connecting to this host	
* tmpFileCreationRetries	: INTEGER = 1000		Number of times Overthere attempts to create a temporary file with a unique name	
* winrmContext	: STRING = /wsman		Context used by the WinRM server	
* winrmEnvelopSize	: INTEGER = 153600		Envelop size for WinRM messages	
* winrmHttpsCertificateTrustStrategy	: ENUM [STRICT, SELF_SIGNED, ALLOW_ALL] = STRICT		HTTPS certificate trust strategy for WinRM over HTTPS	
* winrmHttpsHostnameVerificationStrategy	: ENUM [STRICT, BROWSER_COMPATIBLE, ALLOW_ALL] = STRICT		HTTPS host name verification strategy for WinRM over HTTPS	
* winrmLocale	: STRING = en-US		Locale to use for WinRM messages	
* winrmTimeout	: STRING = PT60.000S		Timeout to use for WinRM messages in XML schema duration format	
tmpDeleteOnDisconnect	: BOOLEAN = true		Whether to delete the temporary connection directory when the connection is closed	
winrmKerberosAddPortToSpn	: BOOLEAN = false		Add the port number (e.g. 5985) to the service principal name (SPN) for which a Kerberos ticket is requested	
winrmKerberosDebug	: BOOLEAN = false		Enable Kerberos debug messages	
winrmKerberosUseHttpSpn	: BOOLEAN = false		Use the HTTP protocol in the service principal name (SPN) for which a Kerberos ticket is requested, instead of the default WSMAN protocol	
Control task	Parameter	CI	Attributes	Description
checkConnection				Checks whether Deployit can transfer files to and execute commands on this host.
destroy			delegate = destroyHost	Shut down EC2 instance and remove all related CIs

cloud.CloudEnvironmentParameters

Type Hierarchy udm.Parameters >> udm.BaseConfigurationItem
Interfaces udm.ConfigurationItem

Parameters for cloud environment instantiation

Public Properties	
* environmentId	: STRING = Environments Id of the environment you want to create
* hostsPath	: STRING = Infrastructure Repository location where all created hosts will appear

cloud.Environment

Type Hierarchy udm.Environment >> udm.BaseConfigurationItem
Interfaces udm.ConfigurationItem

Cloud environment

Public Properties			
* linkedCis	: SET_OF_CI<udm.ConfigurationItem>		CIs that were described in the template of this cloud environment and created along with it
* template	: CI<cloud.EnvironmentTemplate >		Template which was used to create this environment
dictionaries	: LIST_OF_CI<udm.Dictionary>		The dictionaries providing placeholder values. If the same entry exists in multiple dictionaries, the first one in the list is taken.
members	: SET_OF_CI<udm.Container>		The infrastructure components of this Environment
smtpServer	: CI<mail.SmtServer>		The SMTP server used to send mails with when deploying to this Environment.
Control task	Parameter CI	Attributes	Description
destroy			Shut down all related cloud instances and remove all related CIs

cloud.EnvironmentTemplate

Interfaces udm.ConfigurationItem

Cloud environment template

Public Properties			
* hostTemplates	: LIST_OF_CI<cloud.BaseHostTemplate >		Host templates
* xmlDescriptor	: STRING		Freemarker template of XML which describes environment
description	: STRING		Description of the template
Control task	Parameter CI	Attributes	Description
instantiate	cloud.CloudEnvironmentParameters		Instantiate environment and all hosts, which templates are linked to this environment template
validateEnvironmentDescriptor			Validate XML descriptor of the environment template

cloud.HostParameters

Type Hierarchy udm.Parameters >> udm.BaseConfigurationItem

Interfaces udm.ConfigurationItem

Parameters for host templates instantiation

Public Properties			
* hostsLocation	: STRING = Infrastructure		Repository location where all created hosts will appear
instanceName	: STRING		Name of the instance after creation

cloud.SshHost

Type Hierarchy overthere.SshHost >> overthere.RemoteHost >> overthere.Host >> udm.BaseContainer >> udm.BaseConfigurationItem

Interfaces udm.Taggable, udm.ConfigurationItem, udm.Container, overthere.HostContainer

Cloud host with SSH access

Public Properties	
* address : STRING	Address of the host
* cloudId : STRING	Unique ID within cloud platform
* connectionType : ENUM [SFTP, SFTP_CYGWIN, SFTP_WINSSHD, SCP, SUDO, INTERACTIVE_SUDO, TUNNEL] = SFTP	Type of SSH connection to create
* os : ENUM [WINDOWS, UNIX]	Operating system the host runs
* port : INTEGER = 22	Port on which the SSH server runs
* template : CI <cloud.BaseHostTemplate >	Template which was used to create this host
* username : STRING	Username to connect with
jumpstation : CI <overthere.Jumpstation>	Jumpstation that should be used to reach this host
passphrase : STRING	Optional passphrase for the private key in the private key file
password : STRING	Password to use for authentication
privateKeyFile : STRING	Private key file to use for authentication
sudoUsername : STRING	Username to sudo to when accessing files or executing commands
tags : SET_OF_STRING	If set, only deployables with the same tag will be automatically mapped to this container.
temporaryDirectoryPath : STRING	Directory into which temporary files are stored. Will be cleaned up when the connection is closed.

Hidden Properties			
* connectionTimeoutMillis	: INTEGER	= 1200000	Number of milliseconds Overthere waits for a connection to a remote host to be established
* interactiveKeyboardAuthRegex	: STRING	= .*Password:[]?	Regular expression to look for in keyboard-interactive authentication before sending the password
* protocol	: STRING	= ssh	Protocol to use when connecting to this host
* sudoCommandPrefix	: STRING	= sudo -u {0}	Sudo command to prefix to the original command. The placeholder {0} is replaced with the sudoUsername
* sudoPasswordPromptRegex	: STRING	= .*[Pp]assword.*:	Regular expression to look for in interactive sudo before sending the password
* tmpFileCreationRetries	: INTEGER	= 1000	Number of times Overthere attempts to create a temporary file with a unique name
allocateDefaultPty	: BOOLEAN	= false	If true, a default PTY (dummy:80:24:0:0) is allocated when executing a command
allocatePty	: STRING		Specification for the PTY to be allocated when executing a command. The format is TERM:COLS:ROWS:WIDTH:HEIGHT, e.g. xterm:80:24:0:0
sudoOverrideUmask	: BOOLEAN	= true	If true, permissions are explicitly changed with chmod -R go+rX after uploading a file or directory
sudoPreserveAttributesOnCopyFromTempFile	: BOOLEAN	= true	If true, files are copied from the connection temporary directory using the -p flag to the cp command
sudoPreserveAttributesOnCopyToTempFile	: BOOLEAN	= true	If true, files are copied to the connection temporary directory using the -p flag to the cp command
sudoQuoteCommand	: BOOLEAN	= false	If true, the original command is quoted when it is prefixed with sudoCommandPrefix
tmpDeleteOnDisconnect	: BOOLEAN	= true	Whether to delete the temporary connection directory when the connection is closed
Control task	Parameter CI	Attributes	Description
checkConnection			Checks whether Deployit can transfer files to and execute commands on this host.
destroy		delegate = destroyHost	Shut down EC2 instance and remove all related CIs

ec2.Credentials

Interfaces udm.ConfigurationItem

Amazon EC2 credentials

Public Properties			
* key	: STRING		AWS key
* secret	: STRING		AWS secret
Control task	Parameter CI	Attributes	Description
validateCredentials			Make a test call to EC2 with given credentials

ec2.HostTemplate

Type Hierarchy [cloud.BaseHostTemplate](#) >> udm.BaseConfigurationItem

Interfaces udm.ConfigurationItem

Amazon EC2 instance template

Public Properties			
* ami : STRING	The Amazon Machine Image ID (AMI) to use when creating an instance.		
* bootTimeout : INTEGER = 500	Maximal amount of time (in seconds) allowed to elapse before the instance is ready.		
* credentials : CI<ec2.Credentials >	The EC2 credentials to use when communicating with the AWS API.		
* groups : SET_OF_STRING	The EC2 security groups to associate with the instance.		
* instanceType : ENUM [T1Micro, M1Small, M1Medium, M1Large, M1Xlarge, M2Xlarge, M22xlarge, M24xlarge, M3Xlarge, M32xlarge, C1Medium, C1Xlarge, Hi14xlarge, Cc14xlarge, Cc28xlarge, Cg14xlarge]	The EC2 instance type to create.		
* keyPair : STRING	Name of the EC2 key pair to use. It should be already present on EC2.		
* region : STRING	The EC2 region to create the instance in (e.g., us-west-1 or eu-west-1)		
* xmlDescriptor : STRING	Freemarker template of XML which describes instance and middleware		
connectionType : ENUM [SFTP, SFTP_CYGWIN, SFTP_WINSSHD, SCP, SUDO, INTERACTIVE_SUDO, TUNNEL, TELNET, WINRM, WINRM_HTTP, WINRM_HTTPS]	Connection type to be used for connecting to the host		
markerPath : STRING	Path to the file which should appear on the instance when provisioning completes.		
os : ENUM [WINDOWS, UNIX]	OS family		
password : STRING	Password		
privateKeyFile : STRING	Private key file to use for authentication		
retryDelay : INTEGER = 5	Delay (in seconds) after each connection attempt.		
username : STRING	Username		
Control task	Parameter CI	Attributes	Description
instantiate	cloud.HostParameters		Create instance from template
validateDescriptor			Validate XML descriptor