



UrbanCode Deploy plugin for Venafi Trust Protection Platform

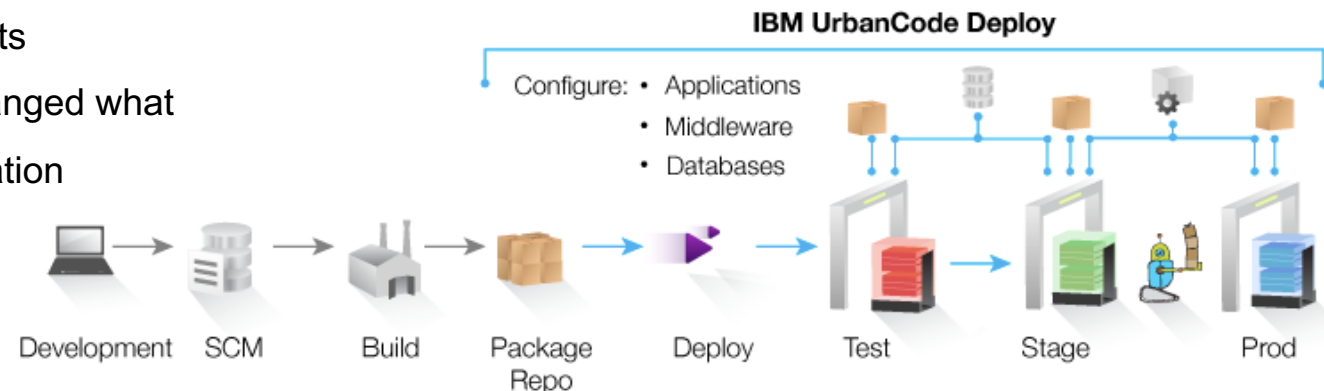
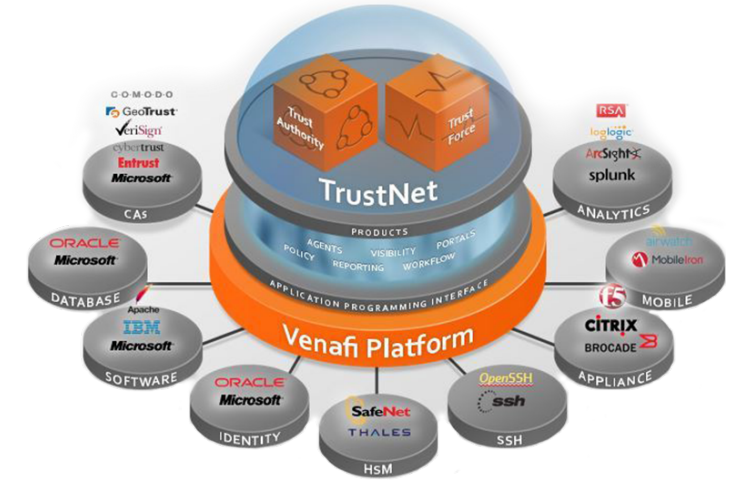
Mark Roberts

UrbanCode Technical Specialist

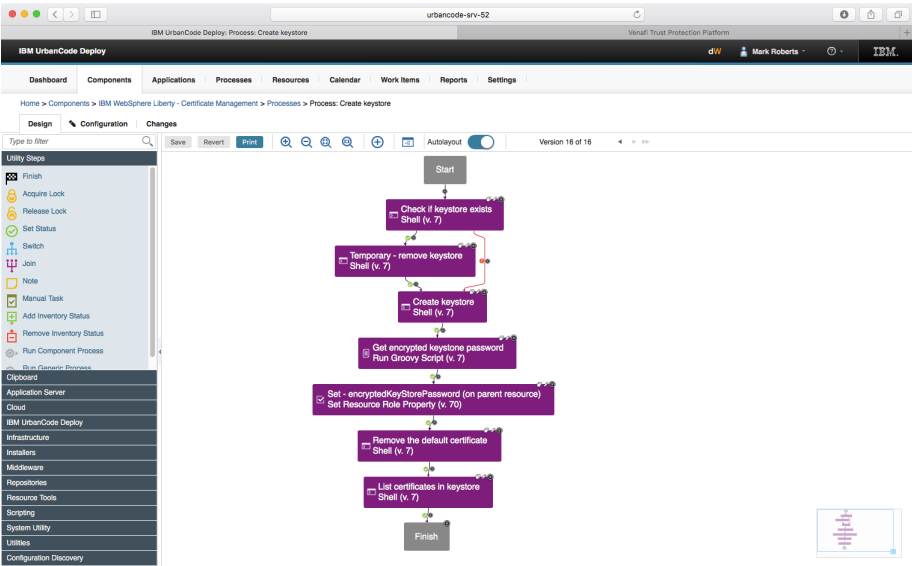
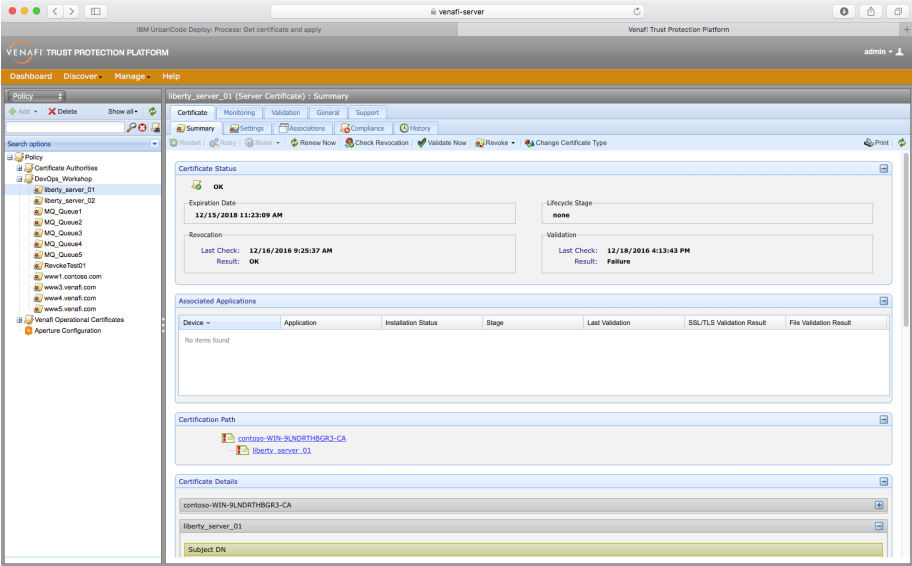
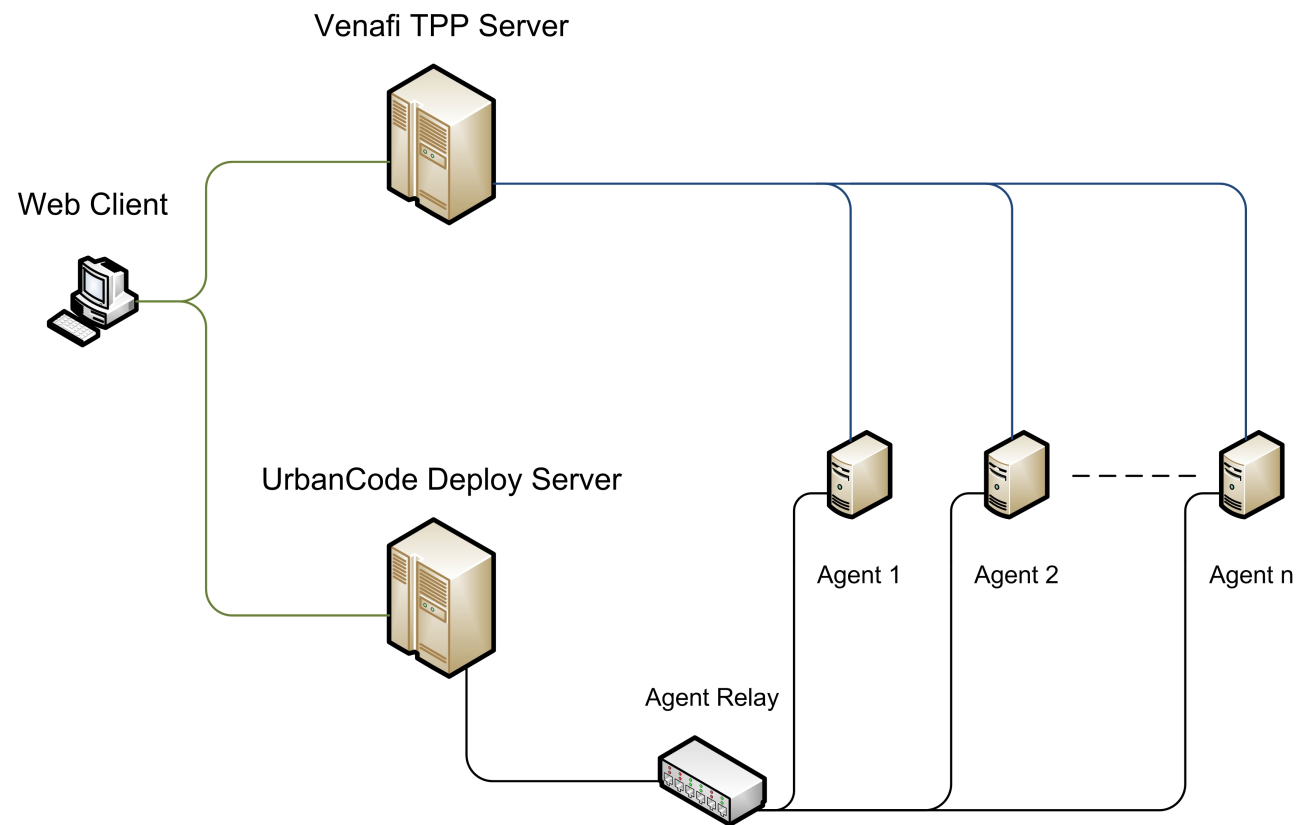
December, 2016

Venafi Trust Protection Platform & UrbanCode Deploy

- Venafi Trust Protection Platform provides
 - Visibility and control over keys and certificates
 - Management of keys and certificates across mobile, desktops, application servers, devices
 - An ever-evolving intelligent response
 - Enforce security policy
 - Continuous monitoring of keys and certificates
- IBM UrbanCode Deploy provides
 - Automated, consistent deployments and rollbacks of application
 - Automated provisioning, updating, and de-provisioning of cloud environments
 - A plugin framework for technology specific extensions – Over 180 plugins available
 - Orchestration of changes across servers, tiers and components
 - Clear inventory visibility: what is deployed where and who changed what
 - Integrated with middleware, provisioning and service virtualization
 - Configuration and security differences across environments
 - Underpin your DevOps strategy



Physical Infrastructure and Interaction



UrbanCode Deploy Plugins

- Over 180 plugins available
- Provided by IBM, UrbanCode user community, partner organisations, IBM specialists
- Written in Groovy
- Interact with third parties using direct API's, REST interfaces, HTTPS communication
- Customers are encouraged to create plugins for their specific needs
- Development process
 - Create standard plugin file and directory structure
 - Add technology specific libraries and interfaces
 - Create test harness scripts and simulated input data files
 - Validate plugin code
 - Create plugin.xml file to connect plugin code to the UrbanCode Deploy user interface to create steps

Example UrbanCode Deploy Plugin Step

- Authentication test step
- Input properties:
 - Venafi TPP server name
 - Venafi TPP user name
 - Venafi TPP user password

Groovy script : authenticationTest.groovy

Creates instance of Groovy class VenafiHelper()

Calls VenafiHelper.authenticate() function

Creates instance of Groovy class VenafiRESTAPI()

Calls VenafiRESTAPI.authenticate() function

High level function for plugin to call

Main plugin code to create messages
and interpret responses

Functions to interface directly with Venafi server

Functions available in the Venafi plugin

- Authentication test – Validate communications with the Venafi TPP server
- Request certificate – Request a certificate from the Venafi TPP. The certificate is not delivered as a part of this step
- Retrieve certificate – Retrieve a certificate from the Venafi TPP server
- Request certificate wait – Request a certificate and wait for it to be returned by the Venafi TPP server
- Get certificate status – Gather information about a specific certificate
- Validate remaining days – Validate whether a certificate will still be valid after a specific number of days
- Revoke certificate – Revoke a certificate on the Venafi TPP server
- Renew certificate – Renew a certificate on the Venafi TPP server. The certificate is not delivered as a part of this step

Request certificate

- Request a new certificate is created
- Certificate is not downloaded at this time
- Provide
 - URL to the TPP Server and username / password
 - Certificate authority distinguished name
 - X.509 Subject (server or service for which the certificate is required)
- Output

Output Properties - [View Input Properties](#)

Name ▲	Value
certificateDN	\\VED\Policy\NewServer
exitCode	0
LOI	
Status	Success
x509Subject	NewServer

Edit Properties

Name * Request Certificate

TPP API URL * \${p:resource/VenafiTPP-URL}

CA DN * \\VED\Policy\Certificate Authorities\MS CA

X.509 subject * NewServer

Working Directory

Post Processing Script Step Default
 [New](#)

Precondition 1

Use Impersonation ☐

Show Hidden Properties ☐

[OK](#)[Cancel](#)

Retrieve certificate

- Retrieve an existing certificate from the Venafi TPP server
- Option to download the certificate chain and the private key
- Provide
 - URL to the TPP Server and username / password
 - Certificate authority distinguished name
 - Filename – Note that the extension is based on certificate format
 - Format – currently only P12
 - Password - A password to protect the downloaded package
- Output
 - Includes the filename for reference by further steps
 - Includes status to show successful retrieval

Edit Properties

Name *	Retrieve Certificate
TPP API URL *	\${p:resource/VenafiTPP-URL}
Certificate DN *	\\VED\\Policy\\DevOps_Workshop\\NewServer
Filename *	NewServerCert
Format *	PKCS #12
Include chain	<input checked="" type="checkbox"/>
Include private key	<input checked="" type="checkbox"/>
Password *
Working Directory	
Post Processing Script	Step Default
	New
Precondition	1
Use Impersonation	<input type="checkbox"/>
Show Hidden Properties	<input type="checkbox"/>

OK

Cancel

Request certificate - Wait

- Request a certificate from the Venafi TPP server and wait for it to be available
- Option to download the certificate chain and the private key
- Provide
 - URL to the TPP Server and username / password
 - Certificate authority distinguished name
 - Policy distinguished name – folder in TPP to hold the certificate
 - Filename – Note that the extension is based on certificate format
 - Format – currently only P12
 - Password - A password to protect the downloaded package
 - Poll time (wait between tries to get certificate)
 - Poll repeats (number of times to ask)
- Output
 - Includes the filename for reference by further steps
 - Includes status to show successful retrieval

Edit Properties

Name *	<input type="text" value="Request Certificate Wait"/>
TPP API URL *	<input type="text" value="https://venafi-server"/>
TPP policy DN *	<input type="text" value="\\VED\\Policy\\DevOps_Workshop"/>
CA DN *	<input type="text" value="\\VED\\Policy\\Certificate Authorities\\MS CA"/>
X.509 subject *	<input type="text" value="liberty_server_01"/>
Poll time *	<input type="text" value="3"/>
Poll repeats *	<input type="text" value="20"/>
Format *	<input type="text" value="PKCS #12"/>
Include chain	<input checked="" type="checkbox"/>
Include private key	<input checked="" type="checkbox"/>
Password *	<input type="password" value="...."/>
Working Directory	<input type="text"/>
Post Processing Script	<input type="text" value="Step Default"/> <input type="button" value="New"/>
Precondition	<div>1</div> <input type="text"/>
Use Impersonation	<input type="checkbox"/>
Show Hidden Properties	<input type="checkbox"/>

OK

Cancel

Get certificate status

- Gather information about a certificate
- Provide
 - URL to the TPP Server and username / password
 - Certificate distinguished name

Output Properties - [View Input Properties](#)

Name ▲	Value
approverDN	\\VED\Identity\admin
approverGUID	local:{6d81fc0d-502b-4ede-a925-0e77e1e30fc4}
contactDN	\\VED\Identity\admin
contactGUID	local:{6d81fc0d-502b-4ede-a925-0e77e1e30fc4}
daysRemaining	730
exitCode	0
keySize	2048
LOI	
processingStage	N/A
processingStatus	N/A
Ready	true
signatureAlgorithm	sha256RSA
Status	Success
validFor	730
validFrom	2016-12-20T07:22:54.0000000Z
validTo	2018-12-20T07:22:54.0000000Z

Edit Properties

Name *

TPP API URL *

Certificate DN *

Working Directory

Post Processing Script

Step Default ▼

Precondition

1

Use Impersonation ☐

Show Hidden Properties ☐

Validate remaining days

- Validate that a certificate is valid for at least a specific number of days
- Provide
 - URL to the TPP Server and username / password
 - Certificate distinguished name
 - Number of days for which the certificate should be valid
 - Dremaining is passed as a component process property in this example
- Step result
 - Step will fail if it cannot connect to Venafi server or find the certificate
 - Step will pass and report validity in 'CertificateOK'
- Output

Output Properties - [View Input Properties](#)

Name ▲	Value
CertificateOK	false
daysRemaining	729

Edit Properties ✕

Name *

Validate Remaining Days

TPP API URL *

\${p:resource/VenafiTPP-URL}

Certificate DN *

\${p:resource/CertificateDN}

Days required *

\${p:DaysRemaining}

Working Directory

Post Processing Script

Step Default ▾
New

Precondition

1

Use Impersonation

☐

Show Hidden Properties

☐

OK

Cancel

Revoke certificate

- Revoke a certificate on the Venafi TPP server
- Provide
 - URL to the TPP Server and username / password
 - Certificate distinguished name
 - Reason :
 - 1 – User key compromised
 - 2 – CA key compromised
 - 3 – User changed affiliation
 - 4 – Certificate superseded
 - 5 – Original use no longer valid
 - Comment – Optional component process property in example

▪ Step result

Output Properties - [View Input Properties](#)

Name ▲	Value
exitCode	0
LOI	
RevokeStatus	true
Status	Success

Edit Properties

Name *

Revoke Certificate

TPP API URL *

\$(p:resource/VenafiTPP-URL}

Certificate DN *

\$(p:resource/CertificateDN}

Reason *

1 - User key compromised ▼

Comment

1 \$(p?:comment}

disabled

☐

Working Directory

Post Processing Script

Step Default ▼
New

Precondition

1

Use Impersonation

☐

Show Hidden Properties

☐

OK

Cancel

Renew certificate

- Renew a certificate on the Venafi TPP server
- Does not necessarily have to be a revoked certificate
- Does not download the certificate – need a retrieve certificate step
- Provide
 - URL to the TPP Server and username / password
 - Certificate distinguished name
- Step result

Output Properties - [View Input Properties](#)

Name ▲	Value
exitCode	0
LOI	
RenewalStatus	true
Status	Success

Edit Properties

Name *

TPP API URL *

Certificate DN *

Working Directory

Post Processing Script

Step Default ▼
New

Precondition

1

Use Impersonation ☐

Show Hidden Properties ☐

OK**Cancel**

Venafi TPP User and password management

- All plugin steps require a login to the Venafi TPP server
- All requests for certificate actions are audited against each user
- Each plugin step requires Venafi username and password
 - tpp Username and tpp User password fields
 - Password field is a secure property
- Username and password values can be:
 - Typed into each step – not ideal as passwords change and would require one ‘generic’ user
 - Stored in the UrbanCode resource tree – not ideal from a security position
 - Typed in for each process execution – Not stored in UrbanCode and each user requesting a deployment must enter their credentials
- Create process properties to request passwords at deploy time

Show Hidden Properties ☒

tpp Username *

tpp User password *

OK Cancel

Note this field value is actually :
\${p:VenafiTPPUserPassword}

Component Process Properties

Add Property

Name	Label	Pattern	Required	Default Value	Description	Actions
VenafiTPPUsername	Venafi User		true	admin	Venafi TPP User name	Edit Delete
VenafiTPPUserPassword	Venafi Password		true	****	Venafi TPP User Password	Edit Delete

2 records - Refresh Print

1 / 1

Rows 10

Venafi TPP User and password management

- At 'deploy' time (execution of a process) the username and password for Venafi TPP are requested
- Password is stored as a secure property and passed to the relevant agents in an encrypted format
- Passwords are redacted in logs
- Each interaction between UrbanCode and Venafi is managed under a specific user ID
- Maintains Venafi audit trails with automation from UrbanCode

Run Process on Development 1 ⓧ

Only Changed Versions ☒

Process * Create Liberty server and deploy application ▼

Select a snapshot, or choose versions for individual components.

Snapshot ▼

Component Versions

Versions 0 selected ([Choose Versions](#))

Request certificate / Venafi User * admin

Request certificate / Venafi Password *

Schedule Deployment? ☐

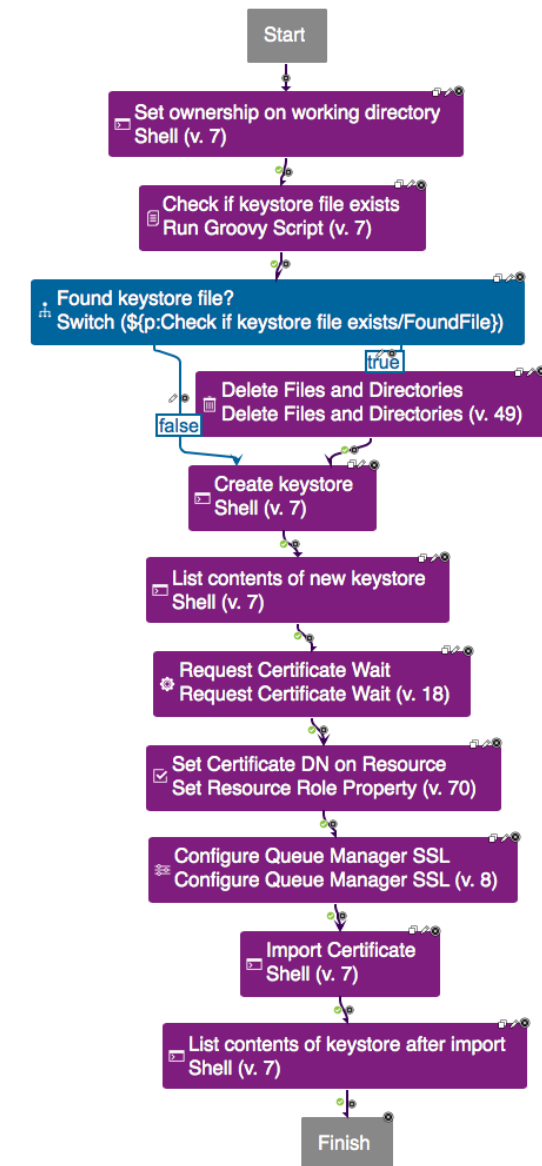
Description

Submit

Cancel

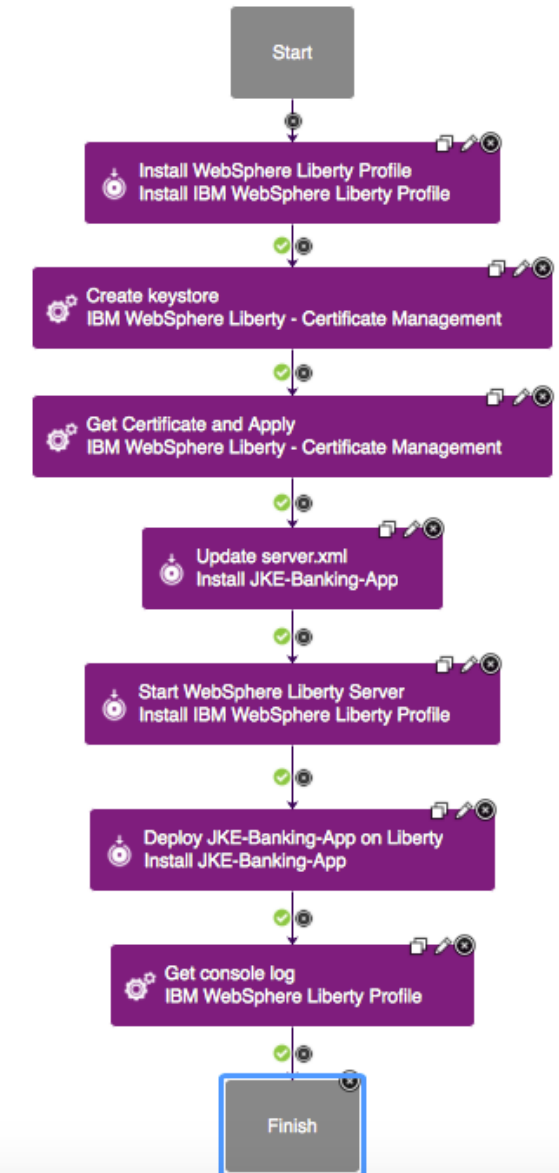
Example process : Request a new certificate and apply to WebSphere MQ

- Complete certificate management process for WebSphere MQ
 - If a keystore is found it is removed and replaced
 - The Certificate DN is stored on the UrbanCode Deploy resource for further interaction with Venafi later such as a certificate validation or renewal
 - Certificates are listed after importing so the user can validate if required
- No need to connect remotely to the target machine



Example process : Install WebSphere Liberty, Certificate and Application

- Install WebSphere Liberty Application Server
- Create a keystore
- Request a certificate and apply to Liberty server
- Update server.xml based on developer requirement for specific features
- Start the Liberty server
- Deploy the application
- Get console log to view status and result of deployment
 - Removes the need to connect to the remote machine



Maintain certificate process

- Check a certificate expiry date in Venafi TPP server records
 - Validate the number of days remaining on the certificate (uses the stored Certificate DN)
 - If the certificate has enough days left then do nothing, otherwise:
 - Request a new certificate on Venafi server
 - Import the new certificate
 - List the new certificate

