

---

# AUDITING WITH INSPEC BADGE TOPICS

The Auditing With InSpec badge is awarded when someone proves that they understand the core elements of InSpec. Candidates must show:

- An understanding of installing inspec.
- An understanding of running inspec.
- An understanding of inspec profiles.
- An understanding of controls and metadata.
- An understanding of troubleshooting InSpec.

Here is a detailed breakdown of each area.

## INSTALLING INSPEC

### INSPEC INSTALLATION AND TARGET NODE

Candidates should understand:

Software InSpec requires on a workstation invoking an InSpec command.

Software InSpec requires on a target.

Installing InSpec on containers.

How InSpec knows what command to run on a target for a particular resource.

### CHEFDK INSPEC OMNIBUS AND RUBYGEM

Candidates should understand:

How to install InSpec software on a workstation.

ChefDK vs. InSpec Omnibus vs. Ruby Gem.

## RUNNING INSPEC

### RUNNING ON A LOCAL SYSTEM

Candidates should understand:

How to invoke InSpec locally.

Using `inspec detect` locally.

Using InSpec to scan the local machine using a local profile.

Using InSpec to scan the local machine using a remote profile.

Invoking InSpec to check a profile contained on your local filesystem.

Invoking InSpec to check a profile stored on a remote server via git.

Invoking specific controls.

### RUNNING ON A REMOTE SYSTEMS VIA SSH

Candidates should understand:

Using `inspec detect` on a remote system.

Using InSpec to scan a local Linux system using a local profile.

---

Using InSpec to scan a local Linux system using a remote profile.  
Using InSpec to scan a remote Linux system using a local profile.  
Using InSpec to scan a remote Linux system using a remote profile.  
Using basic authentication to authenticate with a target system over SSH.  
Using keys to authenticate with a target system over SSH.  
Executing InSpec on a remote Linux system from the command line  
Outputting InSpec scan results as JSON.

#### RUNNING ON REMOTE SYSTEMS VIA WINRM

Candidates should understand:

Using InSpec to scan a local Windows system using a local profile.  
Using InSpec to scan a local Windows system using a remote profile.  
Using InSpec to scan a remote Windows system using a local profile.  
Using InSpec to scan a remote Windows system using a remote profile.  
Authenticating with a target system over WinRM.

#### RUNNING ON CONTAINERS

Candidates should understand:

Using InSpec to scan a docker container using a local profile.  
Using InSpec to scan a docker container using a remote profile.  
Authenticating with a docker container with InSpec.

#### EXAMINING AN API ENDPOINT

Candidates should understand:

The protocols you'd use to test a remote target  
What resource would you use to test URLs  
Implementing resource packs – e.g. AWS, Azure, VMWare.

#### EXAMINING A DATABASE

Candidates should understand:

What SQL database resources are available, and their matchers.  
How to execute an SQL query within a control.

## INSPEC PROFILES

#### METADATA AND THE LOCK FILE

Candidates should understand:

The purpose of the `inspec.yml` file.  
How an InSpec profile's name and version are defined.  
How to define profile dependencies.  
What it means to 'vendor' profiles.  
What the 'inspec.lock' file is.  
Updating the 'inspec.lock' file.  
Setting profile metadata.  
Managing updates from upstream profiles.

---

## PROFILE VERSION SEMVER AND CONSTRAINTS

Candidates should understand:

- The meaning of 'semver' versioning.
- How to set profile version.
- Setting version constraints.
- Setting profile dependencies.

## BASIC RESOURCE TYPES

Candidates should understand:

- Read and understand basic profiles.
- Identify core InSpec resources.
- How to use the common InSpec resources.
- How to find reference documentation on each resource.
- The matchers available for each resource, and when to use each.
- The correct syntax for `describe` statements.
- Matching on `STDERR` and `STDOUT`.
- Using the `file` resource to test directories.

## CREATING CUSTOM RESOURCES

Candidates should understand:

- Why you might want to write a custom InSpec resource.
- Where custom resource code resides.
- Using custom resources within a control.
- Identifying the `InSpec::resource` class.
- Understand the InSpec resource DSL.

## PROFILES INHERITANCE TO OVERLAY CUSTOM CHANGES

Candidates should understand:

- Overlaying custom changes to profiles
- Inheriting only certain controls from another profile.
- Overwriting metadata in inherited controls.

## PROFILES INHERITANCE TO USE CUSTOM RESOURCES

Candidates should understand:

- Using custom resources.
- Using custom matchers.
- How to invoke a custom resource defined in a dependent profile.
- How to invoke a custom resource defined in a dependent resource pack.
- What happens if a custom resource is given the same name as a core resource.

## PROFILE ATTRIBUTES

Candidates should understand:

- Attribute scope – within a control or within a profile.
- Why you would use attributes.
- How and where attribute values are defined.
- How to reference attributes within a control file.

---

Where default values for profile attributes are defined.

#### TESTING EITHER/OR CONFIGURATION CHECKS

Candidates should understand:

When would you use a `describe.one` statement.

When would you use an `only_if` statement.

`describe.one` statement syntax.

`only_if` statement syntax.

When an `only_if` block returns true.

#### SHARING OF PROFILES

Candidates should understand:

How you use the Chef Supermarket.

How you would share a custom resource with the community.

How you would share a custom profile with the community.

How you would run a profile directly from GitHub or the Supermarket.

## CONTROLS AND METADATA

#### USING CONTROL METADATA

Candidates should understand:

What a control's `impact` metadata defines.

What a control's `tag` metadata defines.

What a control's `ref` metadata defines.

The format of a control's tag.

If tag values are arbitrary or a closed set.

If tags can be key value pairs.

Ranking the severity impact of each control using the `impact` metadata, i.e.

0.7 - 1.0 Critical

0.4 - <0.7 Major Issues

0.1 - <0.4 Minor Issues

Using refs to bridge the gap between documented compliance policies and executable code.

## TROUBLESHOOTING

#### USING THE INSPEC SHELL

Candidates should understand:

What InSpec Shell is.

Launching InSpec Shell

Using InSpec Shell.

How you would invoke InSpec shell on a target over SSH

Loading core resources into InSpec Shell.

Loading custom resources into InSpec Shell.