Verifying collections

- Verifying collections with ansible-galaxy
- Verifying signed collections

Verifying collections with `ansible-galaxy`

Once installed, you can verify that the content of the installed collection matches the content of the collection on the server. This feature expects that the collection is installed in one of the configured collection paths and that the collection exists on one of the configured galaxy servers.

```
ansible-galaxy collection verify my_namespace.my_collection
```

The output of the `ansible-galaxy collection verify` command is quiet if it is successful. If a collection has been modified, the altered files are listed under the collection name.

```
ansible-galaxy collection verify my_namespace.my_collection
Collection my_namespace.my_collection contains modified content in the following files:
my_namespace.my_collection
    plugins/inventory/my_inventory.py
    plugins/modules/my_module.py
```

You can use the `-vvv` flag to display additional information, such as the version and path of the installed collection, the URL of the remote collection used for validation, and successful verification output.

```
ansible-galaxy collection verify my_namespace.my_collection -vvv
...
Verifying 'my_namespace.my_collection:1.0.0'.
Installed collection found at '/path/to/ansible_collections/my_namespace/my_collection/'
Remote collection found at 'https://galaxy.ansible.com/download/my_namespace-my_collection-
```

```
1.0.0.tar.gz'
Successfully verified that checksums for 'my_namespace.my_collection:1.0.0' match the remote
collection
```

If you have a pre-release or non-latest version of a collection installed you should include the specific version to verify. If the version is omitted, the installed collection is verified against the latest version available on the server.

```
ansible-galaxy collection verify my_namespace.my_collection:1.0.0
```

In addition to the `namespace.collection_name:version` format, you can provide the collections to verify in a `requirements.yml` file. Dependencies listed in `requirements.yml` are not included in the `verify` process and should be verified separately.

```
ansible-galaxy collection verify -r requirements.yml
```

Verifying against `tar.gz` files is not supported. If your `requirements.yml` contains paths to tar files or URLs for installation, you can use the `--ignore-errors` flag to ensure that all collections using the `namespace.name` format in the file are processed.

Verifying signed collections

If a collection has been signed by a `distribution server`, the server will provide ASCII armored, detached signatures to verify the authenticity of the MANIFEST.json before using it to verify the collection's contents. This option is not available on all distribution servers. See `distributing_collections` for a table listing the servers that support collection signing. See `installing_signed_collections` for how to verify a signed collection when you install it.

To verify a signed installed collection:

```
ansible-galaxy collection verify my_namespace.my_collection  --keyring ~/.ansible/pubring.kbx
```

Use the `--signature` option to verify collection name(s) provided on the CLI with an additional signature. This option can be used multiple times to provide multiple signatures.

```
ansible-galaxy collection verify my_namespace.my_collection --signature
https://examplehost.com/detached_signature.asc --signature
file:///path/to/local/detached_signature.asc --keyring ~/.ansible/pubring.kbx
```

Optionally, you can verify a collection signature with a `requirements.yml` file.

```
ansible-galaxy collection verify -r requirements.yml --keyring ~/.ansible/pubring.kbx
```

When a collection is installed from a distribution server, the signatures provided by the server to verify the collection's authenticity are saved alongside the installed collections. This data is used to verify the internal consistency of the collection without querying the distribution server again when the `--offline` option is provided.

```
ansible-galaxy collection verify my_namespace.my_collection --offline --keyring
~/.ansible/pubring.kbx
```

□□□ #2
□ □□□ □□□ 2026-10-31 14:31:35 UTC
□ □□□ □□□ 2026-10-31 14:31:35 UTC