

The Ansible **k8s** module enables you to manage Kubernetes objects with Ansible playbooks. The **k8s\_auth** module helps manage authentication.

## K8s\_auth parameters

Use the **k8s\_auth** module with clusters requiring explicit authentication procedures (you log in for a token, interact with the API using the token, then log out to revoke).

<code>api_key</code>	Token used to authenticate with an API. When <b>state</b> is set to <b>absent</b> , this specifies the token to revoke.
<code>password</code>	Password for authenticating with an API.
<code>username</code>	Username for authenticating with an API.
<code>validate_certs</code>	Yes → Verify the API server's SSL certificates.
<code>host</code>	The API server. <b>Required</b> .
<code>State</code>	<b>Present</b> → Connect to the API server using the URL specified in <b>host</b> and log in. <b>Absent</b> → Revoke the auth token specified in <b>api_key</b> .

## K8s\_auth example

```
- name: Obtain access token
  k8s_auth:
    username: admin
    password: "{{ k8s_admin_password }}"
    register: k8s_auth_results
- name: Create a namespace with token
  k8s:
    api_key: "{{ k8s_auth_results.k8s_auth.api_key }}"
    name: examplespace
    kind: Namespace
    state: present
```

## K8s parameters

An API set up with HTTP Basic Auth can require an access key. For alternate access methods, such as OAuth2 in OpenShift, use the `k8s_auth` module.

`api_key`           Token to authenticate with an API. Env variable: `K8S_AUTH_API_KEY`

`password`           Password for an API. Env variable: `K8S_AUTH_PASSWORD`

`username`           Username for an API. Env variable: `K8S_AUTH_USERNAME`

Other parameters allow you to perform most actions you could otherwise perform manually with `kubectl`

`host`                URL of the API. Env variable: `K8S_AUTH_HOST`

`src`                 Path to a file containing a valid YAML definition of an object or objects to be created or updated.

`kubeconfig`        Path to an existing Kubernetes config file.

`name`               Specifies an object name when creating, deleting, or discovering an object. Use in conjunction with **`api_version`**, **`kind`**, or **`namespace`** to identify a specific object.

`namespace`         Specifies the namespace to use.

`kind`               Specifies an object model.

`state`             **`Present`** → Create the object.  
**`Absent`** → Delete an existing object.