

## Kube Config

The kubeconfig file is a configuration file used by the Kubernetes command-line tool (kubectl) to authenticate to a Kubernetes cluster and specify various settings for interacting with the cluster. It contains information about the cluster, such as its API server endpoint, authentication credentials, and other settings.

Here's an overview of the key components typically found in a kubeconfig file:

1. **Clusters:** This section defines the Kubernetes clusters you want to interact with. It includes the cluster's name, server URL (API server endpoint), and certificate authority data used to authenticate the server.
2. **Users:** This section defines the users or client certificates used for authentication when accessing the Kubernetes API server. Users can be specified with different authentication methods, such as username/password, client certificates, or token-based authentication.
3. **Contexts:** A context is a combination of a cluster, a user, and optionally a namespace. It specifies the cluster and user to use for a particular operation, along with the default namespace to use if not specified explicitly.
4. **Current Context:** This section specifies the default context to use when running kubectl commands if no context is explicitly provided. It points to a specific context defined in the file.

The kubeconfig file is typically located in the `~/.kube` directory on your local machine, although you can specify its location using the `KUBECONFIG` environment variable. It allows users to switch between different Kubernetes clusters and authentication contexts easily, enabling them to manage multiple clusters from a single command-line interface.

When you run kubectl commands, such as `kubectl get pods`, `kubectl apply`, or `kubectl delete`, the tool reads the kubeconfig file to determine which cluster to communicate with and how to authenticate to it. It's an essential tool for managing Kubernetes clusters and applications.

## Use cases of Kube Config:

The kubeconfig file is a crucial component for interacting with Kubernetes clusters using the kubectl command-line tool. Here are some common use cases for the kubeconfig file:

1. **Accessing Kubernetes Clusters:** The primary purpose of the kubeconfig file is to provide authentication credentials and cluster information, allowing users to securely access Kubernetes clusters. Users can authenticate using various methods, including client certificates, bearer tokens, or username/password combinations.
2. **Managing Multiple Kubernetes Clusters:** Organizations often manage multiple Kubernetes clusters, such as development, staging, and production environments. The kubeconfig file enables users to switch between different clusters seamlessly by configuring multiple cluster entries and contexts within the file.

3. **Role-based Access Control (RBAC):** Kubernetes supports RBAC, allowing administrators to define granular access policies for users and service accounts. The kubeconfig file can specify different users and their associated roles, enabling users to perform specific actions based on their permissions within the cluster.
4. **Multi-tenancy:** In multi-tenant environments, where multiple teams or applications share a single Kubernetes cluster, the kubeconfig file helps enforce isolation by associating users or service accounts with specific namespaces or resource quotas.
5. **CI/CD Pipelines:** Continuous integration and continuous deployment (CI/CD) pipelines often interact with Kubernetes clusters to deploy, update, or scale applications. The kubeconfig file allows CI/CD tools to authenticate with Kubernetes clusters and perform operations such as deploying containerized applications, rolling updates, and monitoring.
6. **Development and Testing:** Developers and testers use the kubeconfig file to interact with Kubernetes clusters during application development and testing phases. They can deploy applications, debug issues, and inspect cluster resources using kubectl commands directly from their development environments.
7. **Automated Scripts and Tools:** Various automation scripts and tools interact with Kubernetes clusters programmatically using the Kubernetes API. These scripts and tools authenticate to the cluster by loading the kubeconfig file, allowing them to perform operations such as scaling deployments, querying cluster state, and managing resources.
8. **Backup and Disaster Recovery:** Backup and disaster recovery processes often involve exporting or backing up the kubeconfig file to ensure that authentication credentials and cluster configurations are preserved. In the event of a disaster or cluster failure, administrators can use the kubeconfig file to restore access to the cluster or migrate workloads to a different cluster.

In this guide, we're delving into the kubeconfig file, a crucial component for interacting with Kubernetes clusters using the kubectl command-line tool. The end goal is to understand the structure and purpose of the kubeconfig file and how it enables users to authenticate to Kubernetes clusters and specify settings for interactions effectively.

### To begin with the Lab:

1. When you first executed your kubectl commands, in few minutes, you must have started wondering how this kubectl is connecting to the Kubernetes cluster. How does it know where is your master node and how does it get authenticated and get all that information from the cluster or even create information in the cluster? Well, the answer is kubeconfig file.
2. When we created the cluster by using kops or by any other method and when we create the Kubernetes cluster, we get a file called kubeconfig file.
3. This file contains all the cluster related information, users, namespaces, and authentication mechanisms.

- Now in your ubuntu session if you do a listing of objects in your home directory.

```
ubuntu@ip-172-31-25-114:~$ ls -a
. . . aws .bash_history .bash_logout .bashrc .cache .kube .profile .ssh .sudo_as_admin_successful .wget-hsts
ubuntu@ip-172-31-25-114:~$ |
```

- And here in this .kube directory you will find a file name config.

```
ubuntu@ip-172-31-25-114:~$ ls .kube/
cache config
ubuntu@ip-172-31-25-114:~$ |
```

- Now if you will open it for that you can write this command.

### **less .kube/config**

- Below is the information which you will find in the kube config file.
- Here you can see that there is written clusters and just below is the certificate for that cluster.

```
apiVersion: v1
clusters:
- cluster:
  certificate-authority-data: LS0tLS1CRUdjTiBDRVJUSUZQ0FURS0tLS0tCk1JSUMrRENDQWVDZ0F3SUJBZ0lNRjhZpRRWtzUWE5bkpzeE1BMEdDU3FHU0l1M0RRRUJd1VBT
UJneEzQVVKQmd0VkJBTVREV3QxWW1WeJtVjBaWE10WFJd0hoY5Na1f3TRjD01UUULnAk13V2hjTk16Ux0RE13TWRreOpNak13V2pBWU1SwXgQVLEVLFRREV3MXJkV0psY201bGRHv
pMV05oTUL1Q0klqU5C2txaGtrz13MeJBuuVGckFBt0NBUTbTbUJ0knS0NBuUWbd1M0bUWuWuNtZGdy0TBlz28r2JRWFWzbUR4c0gwVnRzcDBMz0xjdFmXnmKVU5zFBadkvzY2Z0TFF
Cz1dyd324Mk2NL21zR1LzQ1kZLz0RJN0RCQzRZUk91OD0Yvb1h2djjZuLBxTgdCdkowUpnLcQpbzbdJWluOUVzUWRMKzRGUTRCUWNj1dTeEvbGxjk0JxaX1lzfK1K1ZGTE157Vd0cTCN1LBWMFn
NXdu0XRwCkV1cnf4bxdkexd0TTF2aHBLuZGVjVjeDVvaTVSwVub3E4VfUfTfpkbRzCgcZ09DNGZ6cdwU25dall0a0ckCakxsY09Bu1zBPWUV0d0pwL2VieG18yzZxcfhdj2wUJOYWRCS
0JC0WJBMsYzT3ozWFBU3BYUkttZzVhtk8wegpcis1zzJQN3Fp0it4aUhjbzFZVW02ZmRSU5Nb2tEV0hoK3p6aUtBbXdxJREFRQUjvME13UURBT0JnTLZIUTHCCkFm0EVc0U1d0VfZd0R3WU
RNuJbUQVFU10JBVxDd0VCL3pBZEjntLz1zUtrFRmdRvvdbHzjMWh5dwZmMpVpvY2okWGJ3d1VJdkxZ2E13RFZShtrVkkldm0QVFTeJRQuRnZ0V8CQUg3SFoTjyuuHriTHBLuWLv02FBMFE
5M2hdcpA0YHVF1JhcLNRFvhb1znWgtmXxEyjdNXT3mXc3oNuHjT2ykl1ZHsXkyzzGNEVSvXpCbe91Myt3wZ2SEx0CnNia14u0mUNORFJCK2zMsH9yc1E1RmhoRVZ0OHHmWTFlekyzzFD
NVncGUrbdvJhRDVmNjLczNBZEdkdmUtgkQm1503hsa05au1ktbmM2YTFSZ0pJyvsZFY5dGFVzWlCQjNnVDZY0Fzv2R0HmdhWkz1sUS5NTgxQjNBnNvSTwpMcGrpZEpOUl1znlCuksww
DWDU0FicTlxXbNaCjNQ0mEzRm5tanlpR0tpe17h35lfWc0l0xeBibytBctZWCnjuSTZ0vB4hfYZDjMyyFa1v1JPuutBv1IaHzqdMwQTRIkw2wVRVuMNOs3E3RhGhuRdlhQwRGMstZPQ
otLs0tLu0VRCBDRVJUSUZQ0FURS0tLs0tCk1JSURLeND0WhPz0F3SUJBZ0lSQu04eThlbFJpc3pFwk0yVnB6b2s1Z1v3RFFZSktrWklodmNQ0VFF
server: https://api.cloudservicesdemo.in
  tls-server-name: api.internal.cloudservicesdemo.in
  name: cloudservicesdemo.in
contexts:
- context:
  cluster: cloudservicesdemo.in
  user: cloudservicesdemo.in
  name: cloudservicesdemo.in
current-context: cloudservicesdemo.in
kind: Config
preferences: {}
users:
- name: cloudservicesdemo.in
  user:
    client-certificate-data: LS0tLS1CRUdjTiBDRVJUSUZQ0FURS0tLS0tCk1JSURLeND0WhPz0F3SUJBZ0lSQu04eThlbFJpc3pFwk0yVnB6b2s1Z1v3RFFZSktrWklodmNQ0VFF
TEJQxRXCERFVF01U0dBMV0FQXhNtEVm1wEp1LhsBGN5MwZpZVF1RncweUE5QTBNakf4TtRdd01EVfmGdz85TRkRM8ApNa1370Rd01EVmPfNRE14RnpbVkjnTlZCQW9URG5ONWzUmxx1v
HB0WVh0MFpYSnpnUmN3RLfZRFZRURFdzVycMrXSmxZhlpuTFZw1WRXNTBkVENDQVNjJ0dRnwUpLb1j3ahZjTkfRRUJCUFEZ2dFUEFEQ0NBWu9WDZ2dfQkFNVXEK7G51dGtEakZxclnGRf1NT
ZJa1RDa2VhShd5SkJXUmF2YjcydJxNg5XTBMs01QN0vPsDRFzNhzb3NraUJCSgo2U1Bmcw10U3doazh4VER1Nk17ZwlySGM0Ttg0tEyxc83Vdm2VlRoZEm0ekhXanVjfDgVliYkl
HQ3dtCnTn0SpWnYvUHrb0NFMetGdVhoFg2Tvn5b4tLzs1LerRuxPMENuVpyRenH2NsSzXJ1cEND0Vdc1hJMFEKzndm1nMk5xK3Y0eRf4TG5DvNnkNnUzY4Y0tN
bwY0z1mLsMuM2RkrxMcwtpVjhFZtUzQ0qzyk1aJ6E0c8VzHpkRj1q0LnlwEnLSE1lw0dFQnxB2RufAxQzJd05tTXvNzDfGvC91C1tVjBndhpz3pPzB1RERsW
DBDQxfQUh1ldnR13Rgd2RzfZSMfbkUgqkFRRefnZUfNQk1hQfTzEPru0RtUvR0ndc0dBUVVGQdNQ01Bd0dBMVvRx0F193U0UNNQf3ShZRFZSMgpCQmd3Rm9BvVdEHZjMM5dW
ZmMOpab2NqWgJjd1VJdx2ZE13RFFZSktrWklodmNQ0VFFteJRQuRnZ0VcQUzX0gxFRVFymptQm9aQxdKvk1VeE90ckv0ZJca0Y5u3Zo0WjsNvjaeW81euMcvVtSe14RwlFnkFOV91Rkd
47zFwQ2J6wXRXUDhVVG5ycjN4b1Brdsk0QkVwdXhuZfJrG1wanFnL0twnk5TUfsnjloNmorRzV2lNwM3kcbg9MzLzRNG9yVjFGZxJnM1p0d3FhVTNGTwpwbUtlTETdJx3tc2pmUSt5cHez
L04w0k1DekkZEpNeTdHrzhtbXrwldVkyY0U4Mk1zUo4Nno3aHvFughZxzDckvNL3p6TnlpNHI4bHNGMkwvYmlsQ0RzdGfiuFA1K2EzYDvcC09YcE45VzJra1U3QVRoUmnKZnF1YnV2SEVza
.kube/config
```

- Then you will see there is something called server and its URL. This is the URL of your master node.
- When we created kubernetes cluster using Kops, it did one more thing in Route 53 hosted zone, we see one record that is created by Kops API and its value is the IP address of our master node.

Records (5) DNSSEC signing Hosted zone tags (0)

**Records (5) Info**

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

Record name	Type	Routine...	Differ...	Alias	Value/Route traffic to	TTL (s...)	Health ...
cloudservicesde...	NS	Simple	-	No	ns-885.awsdns-46.net. ns-1172.awsdns-18.org. ns-257.awsdns-32.com. ns-1759.awsdns-27.co.uk.	172800	-
cloudservicesde...	SOA	Simple	-	No	ns-885.awsdns-46.net. awsd...	900	-
api.cloudservices...	A	Simple	-	No	47.129.47.193	60	-
api.internal.clou...	A	Simple	-	No	172.20.61.81	60	-
kops-controller.i...	A	Simple	-	No	172.20.61.81	60	-

**control-plane-ap-southeast-1a.masters.cloudservicesdemo.in i-05dc882694393b763**

**Instance: i-05dc882694393b763 (control-plane-ap-southeast-1a.masters.cloudservicesdemo.in)**

**Details** Status and alarms [New](#) Monitoring Security Networking Storage Tags

**Instance summary** [Info](#)

Instance ID: i-05dc882694393b763 (control-plane-ap-southeast-1a.masters.cloudservicesdemo.in)

Public IPv4 address: 47.129.47.193 [open address](#)

Private IPv4 addresses: 172.20.61.81

```

contexts:
- context:
    cluster: cloudservicesdemo.in
    user: cloudservicesdemo.in
  current-context: cloudservicesdemo.in
kind: Config
preferences: {}
users:
- name: cloudservicesdemo.in
  user:
    client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZQ0FURS0tLS0tCk1JSURLekNDQWhPz0F3SJUBz0LQu04eThbfJpc3pWkeByVnB6b2s1z1v3RFZSkvwklodmNQVFTEJRQxCR0RFV01CU0dBMVVFQxHNTmEzVm1wEp1LhsbNSMwpzVEFLRmcwE5E0TBNakF4TRRd01EVmFnRE14RnpBvkJnT1ZCQW9URG50NWWzUmxxVHB0WhMFPySnpNvMsN3RLfZURFdzvymCmxRSmxZMlpufZwlaRxNTBkVENDQVNJd0RRWUpLb1pJaHjTkFRRUICUUFEEZd2fQkFNVXETKG51dgteAkzxcn1GrJf1NTZ3a1Rda2vhShd5SkJXUm2F2YjcydjzxNg5xTHBMS0lQn08pSDRFZnhZb3NraUC5g02U1BmcwLOU3doazh4VERiNk1TzWlyGm0TTg9TEYxc185Rk83DM2VLRoZEN0eKhXanVjdfdgBvLiYkHQ3dTcJTN085pWnY4VUMrBb0NFMeGdvhvHoFg2TvnSb94TlZsdlErRuXPMENhUpvPyReh2NSszXJcEND0Vdc1hJMFKzmdaM2RCMX0TTU501nMk5xK3Y0eFR4TG5DvNkNuJzYVE4Y0tNbWY021MrSzRjBkxMcWtpVjhVzHpkRj1qQ1NLSTVndENLSeiweDlFQnxFb2R2UFAxQzJd05tTXVNZDFGV9C1c1tWVjbhndHaZ3pPzB1RersWDBDQxFQFHTldNRL1RfZGdZRFZMSFBBuUgqvKFrERFNFzUfNQk1HQTfV2EprUu08KTUfVr0Ndc0dBUVVGQnDQ081Bd0BMVVKRxfDf193JUUNNQF3SHzRfZSMGpC0m3Rm9BVwEdBHZjMM5dWZmQOpab2NqGJJd1Jdkx13RfZkU1p00VFTTEJQRUNrZ0CvQUzxGxtRFFVYmpt0m9aQdxKv1VeE9GckVzQvCa0YU3z0oWJsNvJaae81eUxmCvTsTe14RwfLfnkFovG91Rkd4ZkFWQ2J6WRXzDvHvg5ycjN4b1lBrduUsQkoxHxuFzJrcG1wanFnL01wln5kTfUsnjloNmorRz2V1LnW3kxgb9Mz1zRN9yYfGZJxN1p0d3FhVTngTwppbUtlTetJdXj2cpUst5cHezL04kzNtEkkkzZepNeTfHzRhtbxRwdVkyUuMk1zUu04Nno3aHFGUHzzXkxNL3p6Tn1pHt4bhtbNGMkwvymLSQ0ZrdGfiuFA1kEyZDvCcU9Ycf45VzJra1U3QvRoUrnKZnF1Yn2SEVzamUkCwUtu82MkVnKkHdRDR0enpS5t6zNSTFGStDa3dSedTzmNvWlV1l01DMk3XW9bwB16b1vRwmh3PQotLs0tluv0rcBdvrujsuZj00fURs0tLs0tCgg==client-key-data: LS0tLS1CRUdJTiBSoEgUJjkFURSLRvtLs0tLQpNsULfB3d0kFB50nBUvBeFnvdWQ2MLFPTVxdk1LWfc3bm9pUk1lUjVzKkRjaZaRne5dnZhl3LyFaWRzdWtzcm9nL3tSwzN19GaW1lfrw5StkrkUkx1ENHvhPgtb2b3kNk1tzhZhpnnenc1hxdy9tFzVipodHNwUxCS3RmcjKbs94Uw2Z0lUUW9XNWVGTmzvePhYzfdrWdxQ0uQxUkM3U0upwQm1zTqplL2z1NmtJDFRs2hj1L3Ck0JzDBvNkV3gzcTdkerC4d295W19oUw0CmMy1hsmfVzdxFTst3h3Urxd0RkczNtSkcxhER62X1ysnpd1NoM01rwDfRkv1amt5Me0b0hUsDRR3Ba0IK0dgvVuzaKeyUx0e0nVvlarNzVwlfN0M05tRE02RFJzT0U9WzLJREFRQJzb6b1lcQVFD5v0dL0fEt1nps09jcaowYkd5MgtCtclervd1fVktmekx0WjhCzU1x3cvk1dw1hsdJ5Uz1VQd2hWmRdpv3N3yewHjxWfJycK4z1l0aqRbbk5tMnBw2t1r2pxUTJkW1lx2dTuNh13aZrW1tb3HcL086NtUgM1UzTEVLRFoxU2LpRGRel1E2KfzprbRtNtC9xZSa05R02cxNzdFcK1x1cFNO51nWl01OfVqTkrbzV6yWnqN1l13zRqzJm19rG1Xu0BTTB2Q1pxBxbzmbhhd22zdxNx1nltTtgKbnNsNtLzCzJb8b0dCQVb2ZfZmMGE4Cgw0F0zZEMweUJmRTB0eitRVH1pY01Ns2MvRxy0lUjcfjd2bdXbDVklw3Hfr5Es2etd50FkYndUbxExWdytxak1EWk9jbeX4dwxrK09Wm2RtQ0z1ck9yA2Fub1Zhd2VYMuhaEhiL0ZEctzeHMxQvhb19Ic01afFw2JockpIyUvRn2NPNmQvN2c1bDdMOUey0tKrnBs01QW9QKFnG0K0N9D0VdaMzhyM2RdkRxeKhuKnxvDzMTzHeUs5mtd1ldVzbGbnkdvId1EfEcVpYeVhxz02Tn2t7TM5bE1tqCp2RnA1bEivVuh3aVRhMdhvTloyfRkdMzUxhoga12SHntu0dVsTmVvEh3VwRsg1h1z3zahRrmeyuWVzrjfp0WzNrun11fLzLct1FvbTxExN8yafhJemCa3ZpbEd01kzxWwBb0dBrjFuQ0jdmGUtrk02d0d55j25Gwkb02jcs8vNUZn05c2c3s2lqwm14Vupxzyt0nfNnVedMvnmeEnLYdr1BzL2w2dkf1cgrauJqySw1xegpUtlRwbBscfJyZDlnSmtieG9kMDJwa0xLcGJWn2cSU2y0Vba0FxZxcwR0nMfcZitPeUvKm1pxkV5b2zNc0nFL316eGg0tDzqk0MyU2RfQ245u0NFQ2dZ0pithaJrlWVnm1Kfsm01x3ryNkluaZnjch2Q1jNun0RnsitPnHEksVrnanZLzkxce1v3dZk01u0t01iVlyk3uXezg2nNjK3j3yuZnCtiy0dxtbh2cDfu2ZtjeXnPTdvrMwAapkTge30FBwlwLhbdwdXxV2grcckxpnj3SwUgoyN1cvmlVbe1z05od2hMk3IzbDe5Wdsb1E4VfzXnjFqd1ZwCjkzTnhQuUtC20RSmvUwhxT3dotTRQdE10WfFozVftWff4Q0dVWhZL2pzcTvukWpwQtdeK2L3bHhieDfsbWsKemh5ckRUVx2Vm0wKz1wdG5tby9Zzn1V0WJocjy93eEwxbeGySHF0dG5wYuUhujZURW0xtCs50GhwVt14cnhwZgdDzLbwVkvKvTot5Q1lpdDvQb1VlsdVocmy4Cctod2pXwUy0dWjObWvNSVbqSS2taUN3C10tLs0tRUE5IFJtQ5bQu1wQvrfietFws0tLs0tCg==
```

(END)

14. Now if you will run this command to view your config file then you will see some information but here your certificates are hidden.

```
kubectl config view
```

```
ubuntu@ip-172-31-25-114:~$ kubectl config view
]apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://api.cloudservicesdemo.in
    tls-server-name: api.internal.cloudservicesdemo.in
    name: cloudservicesdemo.in
contexts:
- context:
    cluster: cloudservicesdemo.in
    user: cloudservicesdemo.in
    name: cloudservicesdemo.in
current-context: cloudservicesdemo.in
kind: Config
preferences: {}
users:
- name: cloudservicesdemo.in
  user:
    client-certificate-data: DATA+OMITTED
    client-key-data: DATA+OMITTED
ubuntu@ip-172-31-25-114:~$ ]|
```

15. Now if you want more information about kube config file then you can visit official documentation of Kubernetes.