Installing Standalone Kerberos Server (no FreeIPA)

sudo dnf install krb5-server (client is krb5-workstation, krb5-libs, krb5-user)

/etc/krb5.conf: Main configuration file; defines Kerberos realm, KDC locations, encryption types, etc.
/etc/krb5/kdc.conf: Configuration for the KDC (server-side) if you're setting up a Kerberos server.
/etc/krb5/login.conf: Defines how Kerberos is used for authentication (login).
/var/lib/krb5/krb5.keytab: Stores the master Kerberos key for the KDC.

Client-side commands:
kinit <principal> - For client machine users to get a ticket to access a Kerberos-protected service.
klist -f - Lists all available Kerberos tickets held by the user, for verifying and seeing lifetime, -f gives more info
kdestroy - Destroys a specific Kerberos ticket. For logging out of a service or freeing up resources.
Server-side commands:
kdb5_util - Manages the Kerberos database, keytabs, and principals
kadmin.local - Manages Kerberos principals and credentials: creating, modifying user accounts, resetting passwords, and managing keytabs used by the KDC.

Mentionable related commands/ items:
keyutils - general-purpose tool for managing keyrings and keys, manage Kerberos keytabs alongside other key management tasks. (it itself doesn't interact directly with the Kerberos database)
sshd_krb5_module: This isn't a standalone command, but rather a module used by the SSH daemon to enable Kerberos authentication for SSH connections. You can configure it through SSH configuration files.

Systemd Services:
krb5-kdc.service (server-side): Manages the KDC daemon.
krb5-kadmind.service (server-side): Manages the Kerberos administration daemon.

TCP/88 (default): messages between clients and KDC. TCP is more secure but Windows clients may need UDP

Important Configurations:
Realm: Unique identifier for your Kerberos domain (e.g., EXAMPLE.COM).
KDC Locations: Specify the hostname or IP address of your KDC servers.
Default Encryption Type: Choose an appropriate encryption type (e.g., aes256-cts).
Ticket Lifetime: Set the expiration time for Kerberos tickets.
Client Principal: Define the principal name for your client machine (e.g., host/hostname`).

Managing the Kerberos database, keytabs, and principals with kdb5_util

Create and initialize database and set master password	kdb5_util create -r <realm> -s <keytab_file> -P <passwd>
Create new principal in <realm> with new <password>	kdb5_util addprinc -r <realm> -p <password> <principal>
Modifiy existing principal's attributes (e.g., password, flags)	kdb5_util modifyprinc -r <realm> <principal>
Removes a principal from DB	kdb5_util deleteprinc -r <realm> <principal>
Lists all principals in <realm> with key versions (-kv)	kdb5_util listprinc -r <realm> -kv
Create Keytab	kdb5_util create -r <realm> -s <keytab_file>
Add Entries to Keytab	kdb5_util addprinc -r <realm> -p <passwd> -t <keytab_file> <principal>
Merge Keytabs	kdb5_util merge -s <target_ktab> <source_ktab1> <source_ktab2> ...
Dump Database (can expose sensitive information)	kdb5_util dump -r <realm> -f <output_file>
Verify integrity of the Kerberos database	kdb5_util verify -r <realm>

Manage Kerberos with kadmin.local
Running the command kadmin.local alone will drop you into it's own CLI

Create a new principal for KDC administration:
addprinc -randkey kdc_admin@EXAMPLE.COM
The -randkey option is to generate a random password; kdc_admin@EXAMPLE.COM to name the principal and EXAMPLE.COM representing the Kerberos realm name.

Exit kadmin.local by entering quit.

Grant the kdc_admin principal the permissions to manage the KDC:
kadmin.local -p krb5/admin@EXAMPLE.COM ktadd -k /etc/krb5.keytab kdc_admin@EXAMPLE.COM

The first part "-p krb5/admin@EXAMPLE.COM" provides the password for the krb5/admin principal (usually the root principal) that has full administrative privileges in the Kerberos database.
The second part "ktadd..." adds the key for the kdc_admin principal to the specified keytab file (/etc/krb5.keytab)

Restrict access to kadmin.local using the /etc/sudoers file:
Run "nano /etc/sudoers" and add a block like this:

```
# Allow users in the 'kdc_admin' group to run kadmin.local as kdc_admin@EXAMPLE.COM
%kdc_admin ALL = NOPASSWD: /usr/sbin/kadmin.local -p kdc_admin@EXAMPLE.COM
```

"%kdc_admin" sets the rule applies to users in the kdc_admin group (create it using "groupadd kdc_admin")
"ALL = NOPASSWD" allows group members to run kadmin.local without a password, but only when using the kdc_admin@EXAMPLE.COM principal using the -p option.
"/usr/sbin/kadmin.local -p ..." simply specifies the command with sudo privileges.

Verification:
Create a user account that belongs to the kdc_admin group you created, og in as the newly created user.
Run "sudo kadmin.local -p kdc_admin@EXAMPLE.COM"
You should be prompted for the password of the kdc_admin principal (the one generated in step 2). If successful, you'll enter kadmin.local mode impersonating the kdc_admin principal.

Other Kadmin commands

addprinc <principal>	Adds a new principal (user or service account) to the database
delprinc <principal>	Deletes a principal from the Kerberos database
modprinc <principal>	Modifies attributes of an existing principal
rename_principal <old> <new>	Renames an existing principal in the Kerberos database
change_password <principal>	Changes the password of an existing principal
cpw <principal>	Alias for change_password
listprincs	Lists all principals in the Kerberos database
getprinc <principal>	Retrieves and displays information about a specified principal
ktadd -k <keytab_file> <principal>	Adds a principal's key to a keytab file (for passwordless authentication)
ktremove -k <keytab_file> <principal>	Removes a principal's key from a keytab file
ktdestroy -k <keytab_file>	Destroys a keytab file (use with caution)
getprivs	Shows administrative privileges of current user for kadmin.local CLI
listpols	Lists all policies in database (password rules, ticket lifetimes, etc.)
addpol <policy>	Adds a new policy to the Kerberos database.
modpol <policy>	Modifies attributes of an existing policy.
delpol <policy>	Deletes a policy from the Kerberos database.
getpol <policy>	Retrieves and displays information about a specified policy.
purgekeys <principal>	Removes all keys for a principal that are not the most recent.

SELinux Booleans

allow_httpd_pkey_init	Needed if using HTTP for key distribution.
allow_kadmind_port	Access on TCP 464 for administrative access to the KDC.
allow_kerberos_dce	Needed to support DCE clients using Kerberos.
allow_kerberos_kdc_tcp_port	Enables TCP traffic for the KDC
allow_kerberos_tgt_deleg	Enables delegation of Ticket-Granting Tickets (TGTs)
allow_mit_krb5_migrate	Needed if migrating existing Kerberos principals.
allow_smbd_krb5_right	Required if using Kerberos for Samba authentication.
allow_sshd_klogin	Enables Kerberos login for SSH connections.
allow_unreserved_ports	Allow applications to bind to privileged ports (ports 1-1024)

SELinux File Contexts

/etc/krb5.conf	etc_krb5_conf_t
/var/lib/krb5	var_lib_krb5_t
/var/log/krb5	var_log_krb5_t
Keytab - /etc/krb5.keytab)	krb5_keytab_t
/usr/sbin/kadmin, /usr/sbin/krb5kdc	usr_sbin_krb5_t
/run/krb5 (if used)	var_run_krb5_t

```
firewall-cmd --permanent --add-service=krb5 # Opens default Kerberos ports (TCP 88 and UDP 88)
firewall-cmd --permanent --add-service=kadmind # Opens KDC administration port (TCP 464)

iptables -A INPUT -p tcp --dport 88 -j ACCEPT
iptables -A INPUT -p udp --dport 88 -j ACCEPT
iptables -A INPUT -p tcp --dport 464 -j ACCEPT
systemctl restart krb5kdc kadmin
```

Client configuration - /etc/krb5.conf

```
[libdefaults]
    default_realm = EXAMPLE.COM
    ticket_lifetime = 24h
    renew_lifetime = 7d
[realms]
    EXAMPLE.COM = {
        kdc = kerberos.example.com
        # Optional: Specify additional KDC servers for redundancy
        # kdc = kerberos1.example.com
        # kdc = kerberos2.example.com
    }
[domain_realm]
    .example.com = EXAMPLE.COM
```

Server configuration example /etc/krb5/kdc.conf

```
[kdcdefaults]
    # Define encryption types supported by the KDC
    permitted_enctypes = aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1-96
    default_keytab = /etc/krb5/kdc.keytab
[realms]
    EXAMPLE.COM = {
        # Master key location (use kdb5_passwd to create)
        master_key_file = /var/lib/kerberos/krb5.keytab
        # Database for storing Kerberos principals (replace with your chosen database)
        database_module = kadm5
        # Database specific options
        database_name = EXAMPLE.COM  # Database name for the realm
        # Comment out if database resides on another machine (NOT good to have exposed on the network- don't!)
        # database_server = 192.168.1.10  # Replace with server IP (Not smart! See above)
        # admin_server = kerberos.example.com
        # Restrict access to the KDC based on IP address (Administrative Access Controls are a better option)
        # access_control = {
        #     host = 192.168.1.0/24  # Allow access from this subnet only
        # }
    }
```