Installation Guide

Requirements

- Vault
- Docker
- VaultAIDE code
- Slack workspace with space for a new app
- Nginx
- In case you want TLS on your http Interface you need a valid certificate SSL (not self-signed)

If you don’t have Vault Or Docker installed follow these instructions instead go to Install VaultAIDE

Install Docker

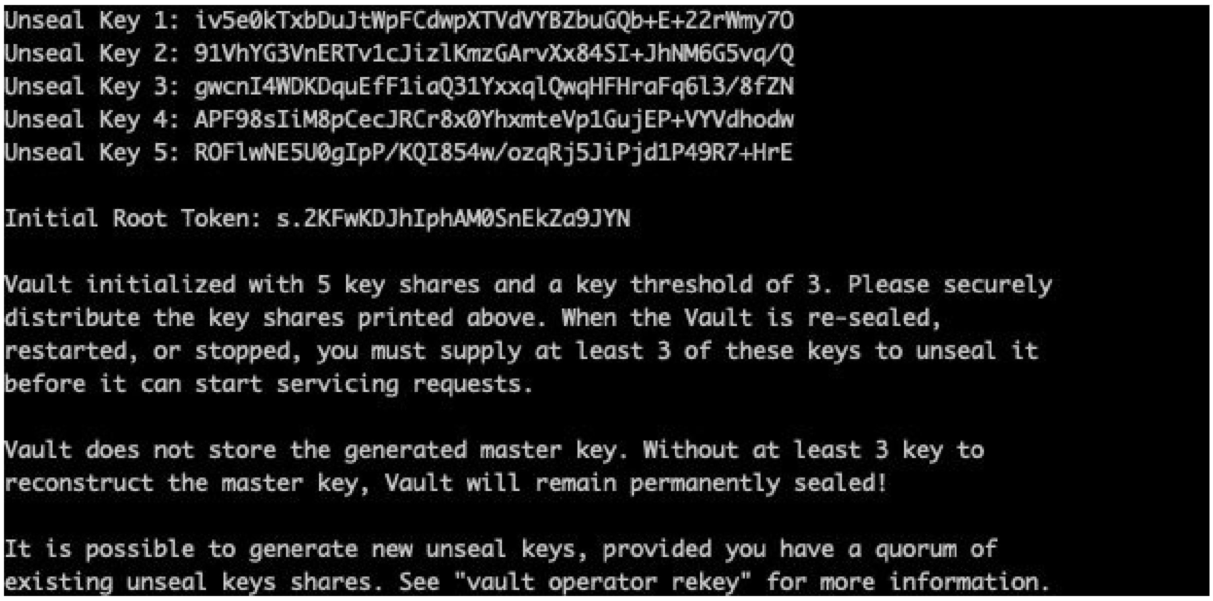
Go to https://docs.docker.com/get-docker/ and follow the steps for your distribution

- For Mac: https://docs.docker.com/docker-for-mac/install/
- For Windows: https://docs.docker.com/docker-for-windows/install/
- For Linux: https://docs.docker.com/engine/install/

Install Vault

Go to https://www.vaultproject.io/downloads and select the button for your operating system and architecture
Then go to https://learn.hashicorp.com/vault/getting-started/install to learn how to install the package
When you’re done you need to go to a terminal and execute this commands:

- *vault operator init*
    - You will see 5 unseal keys and one initial root token



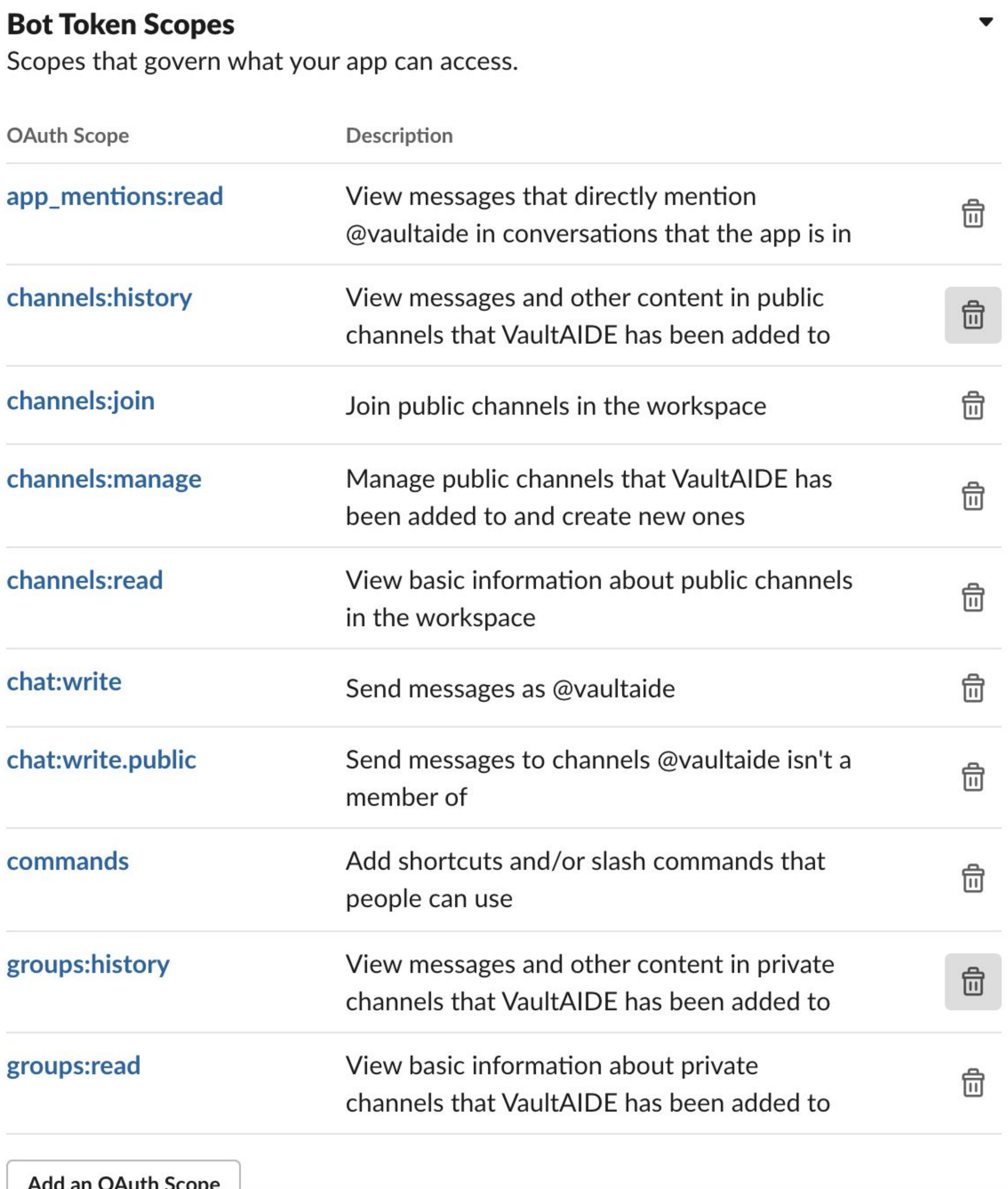
```
Unseal Key 1: iv5e0kTxbDuJtWpFCdwpXTVdVYBZbuGQb+E+22rWmy7O
Unseal Key 2: 91VhYG3VnERTv1cJizlKmzGArvXx84SI+JhNM6G5vq/Q
Unseal Key 3: gwcnI4WDKDquEfF1iaQ31YxxqlQwqHFHraFq6l3/8fZN
Unseal Key 4: APF98sIiM8pCecJRCr8x0YhxmteVp1GujEP+VYVdhodw
Unseal Key 5: ROFlwNE5U0gIpP/KQI854w/ozqRj5JiPjd1P49R7+HrE

Initial Root Token: s.2KFwKDJhIphAM0SnEkZa9JYN

Vault initialized with 5 key shares and a key threshold of 3. Please securely
distribute the key shares printed above. When the Vault is re-sealed,
restarted, or stopped, you must supply at least 3 of these keys to unseal it
before it can start servicing requests.

Vault does not store the generated master key. Without at least 3 key to
reconstruct the master key, Vault will remain permanently sealed!

It is possible to generate new unseal keys, provided you have a quorum of
existing unseal keys shares. See "vault operator rekey" for more information.
```

- *vault operator unseal* (x3)
    - You need to run this command 3 times. On each time you are going to use a different Unseal Key

- *vault audit enable socket address=web:9090 socket_type=tcp*
    - This one is to enable the socket that our bot will be using
- After that you can use your Vault normally and start to experiment with it

Install VaultAIDE

- On a terminal run
  docker pull elizhl/vault-aide
  docker run --rm -p 5000:5000 elizhl/vault-aide
- Point a domain to this project and check that this domain does not have a self-signed certificate (If you are using certificate in your domain the reason that you can't use a self-signed is because slack apps doesn't accept those certificates https://api.slack.com/faq)
- Create an app in slack
    - Go to https://api.slack.com/apps
    - Click *Create new app*
    - Name your app as VaultAide or whatever you want
    - After your app is created go to the Left side Menu -> Features -> OAuth Permissions and select this for bot token scopes

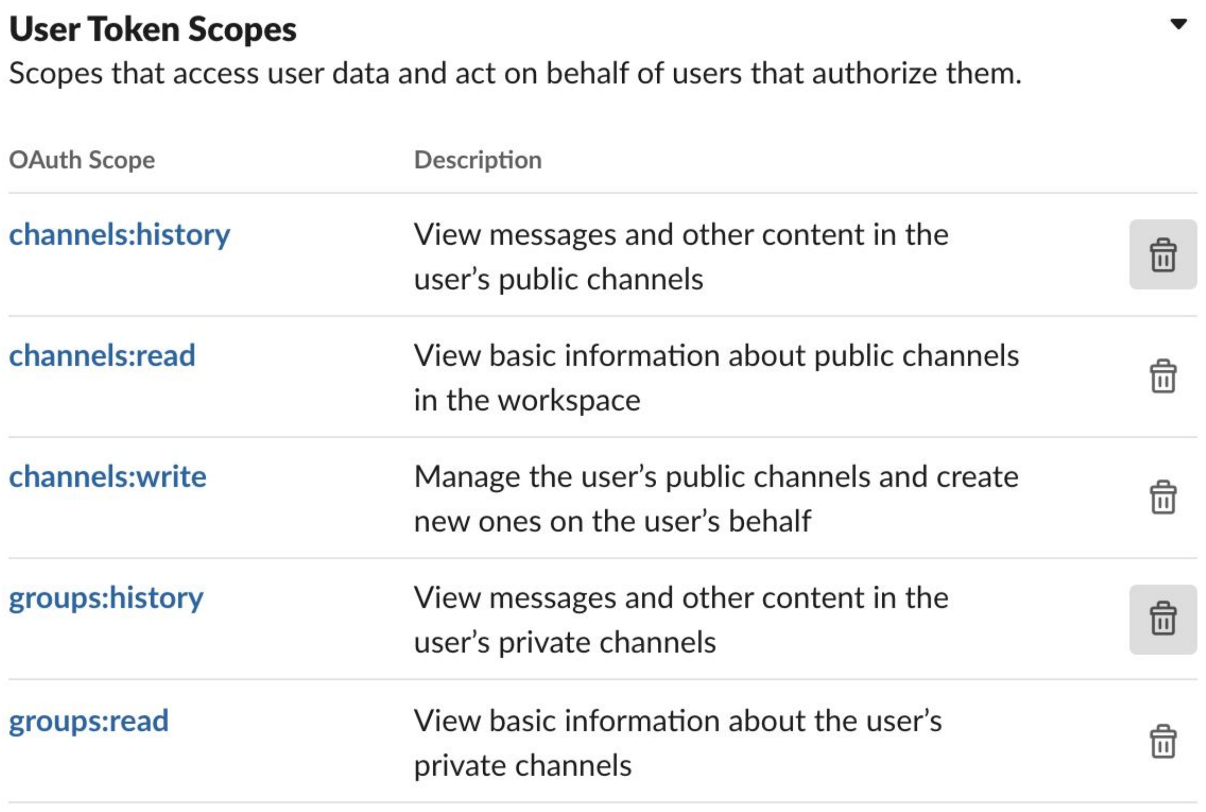


Bot Token Scopes

Scopes that govern what your app can access.

OAuth Scope	Description
app_mentions:read	View messages that directly mention @vaultaide in conversations that the app is in
channels:history	View messages and other content in public channels that VaultAIDE has been added to
channels:join	Join public channels in the workspace
channels:manage	Manage public channels that VaultAIDE has been added to and create new ones
channels:read	View basic information about public channels in the workspace
chat:write	Send messages as @vaultaide
chat:write.public	Send messages to channels @vaultaide isn't a member of
commands	Add shortcuts and/or slash commands that people can use
groups:history	View messages and other content in private channels that VaultAIDE has been added to
groups:read	View basic information about private channels that VaultAIDE has been added to

- and this for user token scopes:

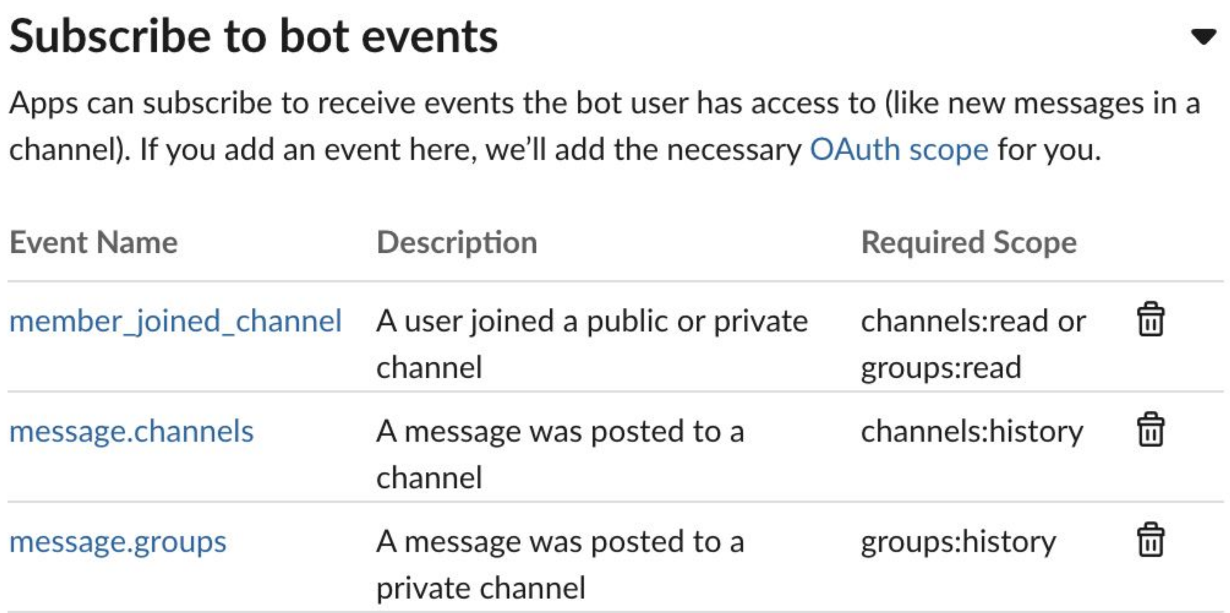


User Token Scopes

Scopes that access user data and act on behalf of users that authorize them.

OAuth Scope	Description
channels:history	View messages and other content in the user's public channels
channels:read	View basic information about public channels in the workspace
channels:write	Manage the user's public channels and create new ones on the user's behalf
groups:history	View messages and other content in the user's private channels
groups:read	View basic information about the user's private channels

- Then on top of the page click *Install your app*
- Copy the "Bot User OAuth Access Token" and make sure to save that token because later we are going to need it.
- Go to the Left side Menu -> Event Subscriptions
- Click the toggle button to turn it on
- A field text will appear for you to put your *domain URL + slack/get-answer. Example: My url is https://8cd73700.ngrok.io so in my Request URL I'm going to*
- *write https://8cd73700.ngrok.io/slack/get-answer* (This is the way slack will be sending notifications to our VaultAide to get Vault information)
- After you get a Verified text in your URL go down and click *Subscribe to bot events* and select this ones

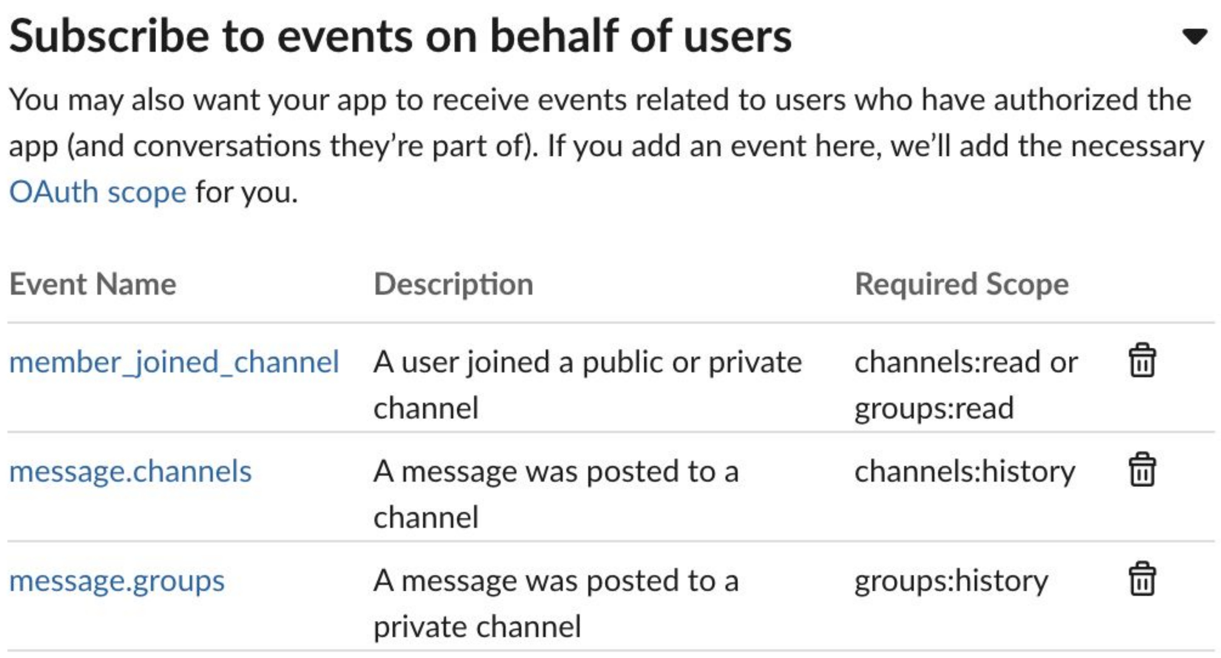


Subscribe to bot events

Apps can subscribe to receive events the bot user has access to (like new messages in a channel). If you add an event here, we'll add the necessary OAuth scope for you.

Event Name	Description	Required Scope
member_joined_channel	A user joined a public or private channel	channels:read or groups:read
message.channels	A message was posted to a channel	channels:history
message.groups	A message was posted to a private channel	groups:history

   - Now click *Subscribe to events on behalf of users* and select this events:



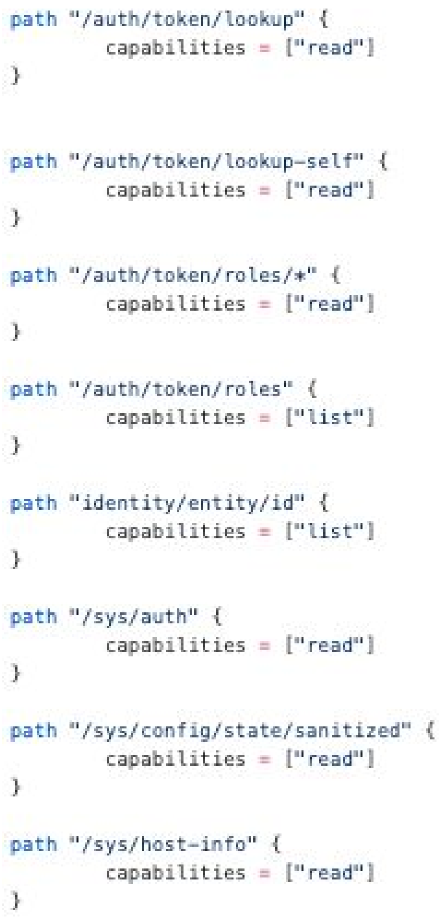
Subscribe to events on behalf of users

You may also want your app to receive events related to users who have authorized the app (and conversations they're part of). If you add an event here, we'll add the necessary OAuth scope for you.

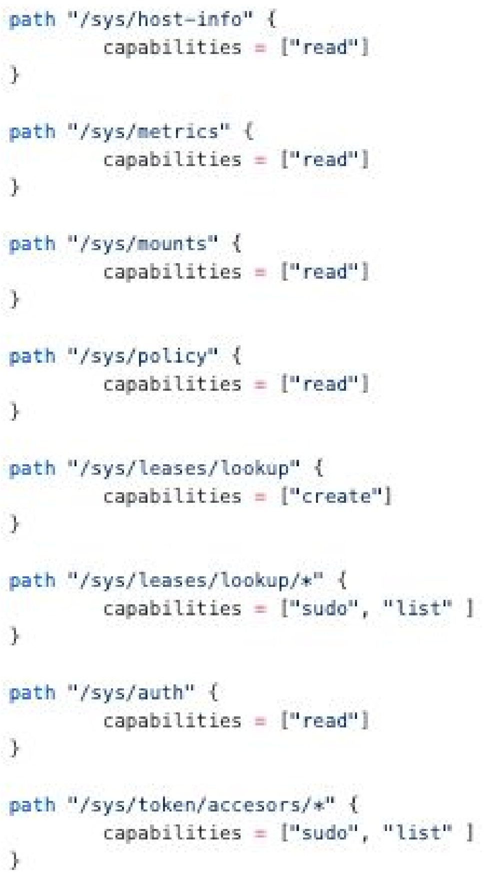
Event Name	Description	Required Scope
member_joined_channel	A user joined a public or private channel	channels:read or groups:read
message.channels	A message was posted to a channel	channels:history
message.groups	A message was posted to a private channel	groups:history

   - Then click *Save Changes* in some cases a message for reinstall your app may appear. You just need to click the text that says *Reinstall your app*

- Go to your workspace in slack and create a channel, you can name this as you like.
- Now go to your Vault and create a token with this policies

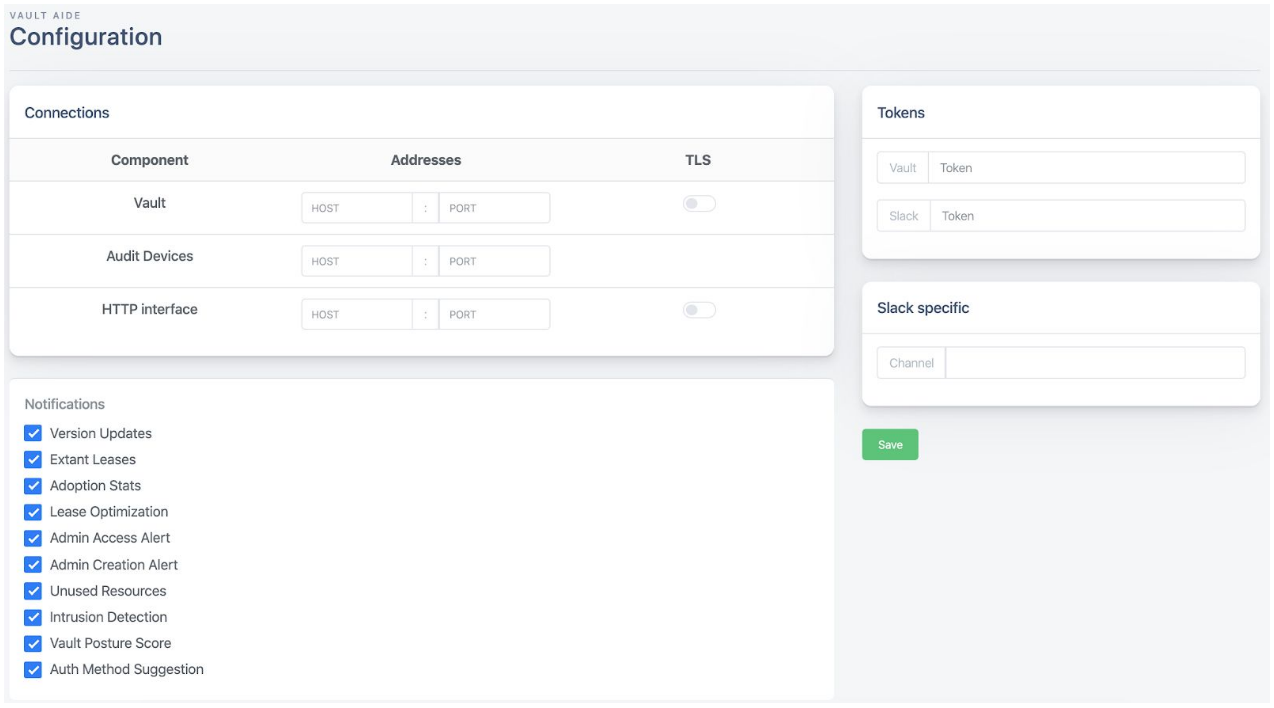


```
path "/auth/token/lookup" {
        capabilities = ["read"]
}

path "/auth/token/lookup-self" {
        capabilities = ["read"]
}

path "/auth/token/roles/*" {
        capabilities = ["read"]
}

path "/auth/token/roles" {
        capabilities = ["list"]
}

path "identity/entity/id" {
        capabilities = ["list"]
}

path "/sys/auth" {
        capabilities = ["read"]
}

path "/sys/config/state/sanitized" {
        capabilities = ["read"]
}

path "/sys/host-info" {
        capabilities = ["read"]
}
```



```
path "/sys/host-info" {
        capabilities = ["read"]
}

path "/sys/metrics" {
        capabilities = ["read"]
}

path "/sys/mounts" {
        capabilities = ["read"]
}

path "/sys/policy" {
        capabilities = ["read"]
}

path "/sys/leases/lookup" {
        capabilities = ["create"]
}

path "/sys/leases/lookup/*" {
        capabilities = ["sudo", "list" ]
}

path "/sys/auth" {
        capabilities = ["read"]
}

path "/sys/token/accesors/*" {
        capabilities = ["sudo", "list" ]
}
```

- Make sure to save that token too
- In a browser like Chrome, go to your *domain URL + / config. Example: My domain as you know is https://8cd73700.ngrok.io so in my browser I'm going to*
- *write https://8cd73700.ngrok.io/config*
- You will see a page like this


VAULT AIDE
Configuration

Connections

Component	Addresses	TLS
Vault	HOST : PORT	○
Audit Devices	HOST : PORT	
HTTP interface	HOST : PORT	○

Tokens

Vault: Token

Slack: Token

Slack specific

Channel:

Save

Notifications

- ✓ Version Updates
- ✓ Extant Leases
- ✓ Adoption Stats
- ✓ Lease Optimization
- ✓ Admin Access Alert
- ✓ Admin Creation Alert
- ✓ Unused Resources
- ✓ Intrusion Detection
- ✓ Vault Posture Score
- ✓ Auth Method Suggestion

- So as you will see it is time to give the information to our application. In this page you are going to save the slack and vault tokens, the slack channel, addresses for Vault, Audit Device and HTTP Interface so let's start with Connections.
  - For Vault connection you need to provide a host and port to know where can we find your instance. At the same time you can let the app know if your Vault has TLS enabled.
  - For Audit Devices you can provide a host and port to know where the log information is available in this case this need to be a socket.
  - For HTTP interface you can provide an action URL to let the app know what's the slack event URL and same as Vault you can tell if this has TLS.
- Then you will see a Notifications section
  - Those are the available notifications that this app can provide so you can select what are useful for your purposes
    - Version Updates will check your vault version and compare with the latest available so in case you don't have the latest this app will let you know
    - Adoption Stats will give you some general information about your Vault
    - Extant Leases will deliver information about the leases like the total leases
    - Lease Optimization will check if you are giving to many ttl to your leases and suggest some changes about it
    - Admin Access Alert will deliver a notification every time that an admin token perform an action in your vault
    - Admin Creation Alert will deliver a notification every time that a root account is created in your Vault.
    - Unused resources will check if you have some secrets engine that you are not using

        - Intrusion Detection will be watching the unwrapping action so the app can let you know when some unwrapping failure is detected
        - Vault Posture Score will be checking some points to let you know how close you are to being "as secure as possible"
        - Auth Method Suggestions will check if you are using secure auth methods and if you’re not then will suggest some upgrades
    - Then you can see the Token section and in this one we need the tokens we create previously Vault and Slack tokens.
    - And finally you see the Slack Specific section that is just for the slack channel. We also created this previously
    - After you populate all the information you can click *Save*
    - Our application will restart and we can wait just for the first notification to appear.
    - In case you don’t want to wait you can type in the slack channel some of this:
        - Adoption Stats
        - Vault Score
        - Version
        - Lease Optimization

Vault Resources

Version Updates:

- HTTP request to /v1/sys/health to get the **version** value (GET)

Extant Leases:

- HTTP request to /v1/sys/leases/lookup/auth/token/create to get the total **leases** (LIST)

Adoption Stats:

- Audit Log entries to get number of **operations** per month or week
- Python client HVAC -> List of **auth methods**, **secrets engine**, **policies**
- HTTP request to /v1/identity/entity/id to get the total **entities** (LIST)
- HTTP request to /v1/auth/token/accessors to get total **tokens** (LIST)
- HTTP request to /sys/auth/roles and /sys/auth/users to get total **roles** (LIST)

Lease Optimization:

- Audit Log entries to get the **ttl** and **time of use** for every lease

Admin Access Alert:

- Audit Log entries to know when a **root token is used**

Admin Creation Alert:

- Audit Log entries to know when a **root token is created**

Unused Resources:

- Audit Log entries to get **resources** with no log entries

Intrusion Detection:

- Audit Log entries to track **unwrapping token fails**

Vault Posture Score:

- We use some of the **previous functions** like
  Version Updates
  Admin Access and Creation Alert
  Python CLient HVAC -> List of Auth methods