Guidelines

In this tutorial, I will guide you how to deploy the Kratos service to Kubernetes. The requirement of infratructure includes

- Kubernetes > v1.22
- Postgres > v13.0 or Mysql > v8.0
- Redis > v6.0
- Both UI & Kratos must be configured under same domain and SSL must be turned on. For example you will have 2 domain `auth.example.com` and `kratos.example.com` for UI and Kratos service respectively

Check lists

- Make sure you have configured sticky session for loadbalancer
- If you are using DNS service (Cloudflare), make sure you use Strict SSL configurion. That mean the connection between your server and Cloudflare must be HTTPS to make cookie-session model works well
- Never, ever expose Kratos Admin interface (port `4434`) to the public internet

Build Docker image

You can use this command to build the Docker image that will be used to deploy to Kubernetes. Make sure you replace `scrapnode/kratos` with your docker image name

```
docker build . -f .docker/Dockerfile-build --tag scrapnode/kratos:latest
```

```
[+] Building 177.9s (21/21) FINISHED
 => [internal] load build definition from Dockerfile-build
 => => transferring dockerfile: 1.34kB
 => [internal] load .dockerignore
 => => transferring context: 203B
 => resolve image config for docker.io/docker/dockerfile:1-experimental
 => CACHED docker-image://docker.io/docker/dockerfile:1-experimental@sha256:600e5c62eedff338
 => [internal] load metadata for gcr.io/distroless/base-nossl-debian12:nonroot
 => [internal] load metadata for docker.io/library/golang:1.21
 => [builder  1/10] FROM docker.io/library/golang:1.21@sha256:1a9d253b11048b1c76b690b0c09d78
 => => resolve docker.io/library/golang:1.21@sha256:1a9d253b11048b1c76b690b0c09d78d200652e4e
 => => sha256:1a9d253b11048b1c76b690b0c09d78d200652e4e913d5d1dcc8eb8d0d932bfe4 2.36kB / 2.36
 => => sha256:e30422593ba0e5ef53b7ad32685a9788f630ccf2357d12a98cbaabfd0e8905db 1.58kB / 1.58
 => => sha256:b5de22c0f5cd2ea2bb6c0524478db95bff5a294c99419ccd4a9d3ccc9873bad9 24.05MB / 24.
 => => sha256:533c8541132bd6cceaa971c458463325521909fbd6262d798c32d3ed30bea435 7.55kB / 7.55
 => => sha256:bc0734b949dcdcabe5bfdf0c8b9f44491e0fce04cb10c9c6e76282b9f6abdf01 49.56MB / 49.
 => => sha256:917ee5330e73737d6095a802333d311648959399ff2c067150890162e720f863 64.13MB / 64.
 => => sha256:bed956b3d08c0b1903d9af3e6850930eacf865b23f62bd8cf46eb1de1325c39b 92.36MB / 92.
 => => extracting sha256:bc0734b949dcdcabe5bfdf0c8b9f44491e0fce04cb10c9c6e76282b9f6abdf01
```

```
 => => sha256:6febcc5ccdb1e72a8b6ce15a28ee29f8e9bd7fc5debbcf94903105c2c2578698 66.98MB / 66.
 => => sha256:51575304560117c73ccac37a74ad8cab516786a9861308fc3e3e0575d0a06c4d 155B / 155B
 => => extracting sha256:b5de22c0f5cd2ea2bb6c0524478db95bff5a294c99419ccd4a9d3ccc9873bad9
 => => extracting sha256:917ee5330e73737d6095a802333d311648959399ff2c067150890162e720f863
 => => extracting sha256:bed956b3d08c0b1903d9af3e6850930eacf865b23f62bd8cf46eb1de1325c39b
 => => extracting sha256:6febcc5ccdb1e72a8b6ce15a28ee29f8e9bd7fc5debbcf94903105c2c2578698
 => => extracting sha256:51575304560117c73ccac37a74ad8cab516786a9861308fc3e3e0575d0a06c4d
 => [runner 1/3] FROM gcr.io/distroless/base-nossl-debian12:nonroot@sha256:8c957f0c06030921e
 => [internal] load build context
 => => transferring context: 9.63MB
 => [builder  2/10] RUN apt-get update && apt-get upgrade -y &&  mkdir -p /var/lib/sqlite
 => [builder  3/10] WORKDIR /go/src/github.com/ory/kratos
 => [builder  4/10] COPY go.mod go.mod
 => [builder  5/10] COPY go.sum go.sum
 => [builder  6/10] COPY internal/httpclient/go.* internal/httpclient/
 => [builder  7/10] COPY internal/client-go/go.* internal/client-go/
 => [builder  8/10] RUN go mod download
 => [builder  9/10] COPY . .
 => [builder 10/10] RUN --mount=type=cache,target=/root/.cache/go-build go build -tags sqlit
 => CACHED [runner 2/3] COPY --from=builder --chown=nonroot:nonroot /var/lib/sqlite /var/lib
 => [runner 3/3] COPY --from=builder --chown=nonroot:nonroot /usr/bin/kratos /usr/bin/kratos
 => exporting to image
 => => exporting layers
 => => writing image sha256:94cef24141fb60735649bb65645d95e2566296bf92e7982d7f49cdf95a7908eb
 => => naming to docker.io/scrapnode/kratos:latest

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to
```

Prepare configurations

You need to prepare 2 kind of configirations:

1. A `ConfigMap` that store the Krato Indentity Schema what define how you will store user data in Kratos system

Kubernetes ConfigMap

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: identity-schemas
  namespace: identity
data:
  default.json: |
    {
      "$id": "kratos/identity/schema/default",
      "$schema": "http://json-schema.org/draft-07/schema#",
```

```
  "title": "Account",
  "type": "object",
  "properties": {
    "traits": {
      "type": "object",
      "properties": {
        "email": {
          "type": "string",
          "format": "email",
          "title": "E-Mail",
          "minLength": 3,
          "ory.sh/kratos": {
            "credentials": {
              "password": {
                "identifier": true
              }
            },
            "verification": {
              "via": "email"
            },
            "recovery": {
              "via": "email"
            }
          }
        }
      },
      "required": ["email"],
      "additionalProperties": false
    }
  }
}
```

2. A `Secret` that store Kratos configuration that contains sensitive data like database connection string, cookie & session secret, ...

There are so many configurations you can set, but there are some important parts you need to change before deploy your Kratos servcie

- `dns`: Database Connection String, we are supporting both PostgreSQL and MySQL database. Examples:
  - PostgreSQL connection string: `postgres://postgres:changemenow@127.0.0.1/postgres?connec`
  - MySql connection string: `root:changemenow@tcp(127.0.0.1:3306)/default?charset=utf8mb4&&`
- Replace all example configuration for UI domain with your UI hosted domain(example: `https://auth.example.com` and any value with `example.com`)
  - `selfservice.default_browser_return_url`

  - `selfservice.flows.logout.after.default_browser_return_url`
  - `selfservice.flows.registration.ui_url`
  - `selfservice.flows.login.ui_url`
  - `selfservice.flows.verification.ui_url`
  - `selfservice.flows.verification.after.default_browser_return_url`
  - `selfservice.flows.recovery.ui_url`
  - `selfservice.flows.error.ui_url`
  - `selfservice.flows.settings.ui_url`
  - `serve.public.cors.allowed_origins`
- Configures outgoing emails using the SMTP protocol by replacing
  - `courier.smtp.connection_uri` with your SMTP connection string
  - `courier.smtp.from_address` with your sender email address
  - `courier.smtp.from_name` with your sender name
  - `courier.smtp.X-SES-FROM-ARN` is required if you are using AWS SES
- Replace Kratos URLs with your configured domains
  - `serve.public.base_url` wiht Kratos public domain (example: `https://kratos.example.com/`)
  - `serve.admin.base_url` wiht Kratos admin domain (example: `http://kratos.svc.cluster.local:4434/`)
- You also may want to change some security secrets of
- `secrets.cookie`
- `secrets.cipher`
- IMPORANT: `cookies.domain` must be set with your root domain (DON'T FORGET THE BEGINNING DOT). For example `.example.com`. Read this article to get more information about cookie-session domain setting (https://stackoverflow.com/questions/18492576/share-cookies-between-subdomain-and-domain)

Kratos Configuration

```
## Ory Kratos Configuration
#

## identity ##
#
identity:

  ## All JSON Schemas for Identity Traits ##
  #
  # Note that identities that used the "default_schema_url" field in older kratos versions u
  #
  # Examples:
```

```
  # - - id: customer
  #     url: base64://ewogICIkc2NoZW1hIjogImh0dHA6Ly9qc29uLXNjaGVtYS5vcmcvZHJhZnQtMDcvc2NoZ
  #   - id: employee
  #     url: https://foo.bar.com/path/to/employee.traits.schema.json
  #   - id: employee-v2
  #     url: file://path/to/employee.v2.traits.schema.json
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export IDENTITY_SCHEMAS=<value>
  # - Windows Command Line (CMD):
  #    > set IDENTITY_SCHEMAS=<value>
  #
  schemas:
    - id: default
      url: file:///identity/schemas/default.json

  ## The default Identity Schema ##
  #
  # This Identity Schema will be used as the default for self-service flows. Its ID needs to
  #
  # Default value: default
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export IDENTITY_DEFAULT_SCHEMA_ID=<value>
  # - Windows Command Line (CMD):
  #    > set IDENTITY_DEFAULT_SCHEMA_ID=<value>
  #
  default_schema_id: default

## Data Source Name ##
#
# DSN is used to specify the database credentials as a connection URI.
#
# Examples:
# - "postgres://user:
#   password@postgresd:5432/database?sslmode=disable&max_conns=20&max_idle_conns=\
#   4"
# - mysql://user:secret@tcp(mysqld:3306)/database?max_conns=20&max_idle_conns=4
# - cockroach://user@cockroachdb:26257/database?sslmode=disable&max_conns=20&max_idle_conns=
# - sqlite:///var/lib/sqlite/db.sqlite?_fk=true&mode=rwc
#
# Set this value using environment variables on
# - Linux/macOS:
#    $ export DSN=<value>
```

```
# - Windows Command Line (CMD):
#    > set DSN=<value>
#
dsn: "postgres://POSTGRES_USER:POSTGRES_PASSOWRD@POSTGRES_HOST:5432/POSTGRES_DATABASE?connec

## selfservice ##
#
selfservice:

  ## Redirect browsers to set URL per default ##
  #
  # Ory Kratos redirects to this URL per default on completion of self-service flows and oth
  #
  # Examples:
  # - https://my-app.com/dashboard
  # - /dashboard
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SELFSERVICE_DEFAULT_BROWSER_RETURN_URL=<value>
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_DEFAULT_BROWSER_RETURN_URL=<value>
  #
  default_browser_return_url:  https://auth.example.com/

  ## flows ##
  #
  flows:

    ## logout ##
    #
    logout:

      ## after ##
      #
      after:

        ## Redirect browsers to set URL per default ##
        #
        # Ory Kratos redirects to this URL per default on completion of self-service flows c
        #
        # Examples:
        # - https://my-app.com/dashboard
        # - /dashboard
        #
        # Set this value using environment variables on
```

```
    # - Linux/macOS:
    #    $ export SELFSERVICE_FLOWS_LOGOUT_AFTER_DEFAULT_BROWSER_RETURN_URL=<value>
    # - Windows Command Line (CMD):
    #    > set SELFSERVICE_FLOWS_LOGOUT_AFTER_DEFAULT_BROWSER_RETURN_URL=<value>
    #
    default_browser_return_url: https://auth.example.com

## registration ##
#
registration:

  ## Registration UI URL ##
  #
  # URL where the Registration UI is hosted. Check the [reference implementation](https
  #
  # Default value: https://www.ory.sh/kratos/docs/fallback/registration
  #
  # Examples:
  # - https://my-app.com/signup
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SELFSERVICE_FLOWS_REGISTRATION_UI_URL=<value>
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_FLOWS_REGISTRATION_UI_URL=<value>
  #
  ui_url: https://auth.example.com/registration

  ## lifespan ##
  #
  # Default value: 1h
  #
  # Examples:
  # - 1h
  # - 1m
  # - 1s
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SELFSERVICE_FLOWS_REGISTRATION_LIFESPAN=<value>
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_FLOWS_REGISTRATION_LIFESPAN=<value>
  #
  lifespan: 1h

  ## after ##
```

```
  #
  after:

    ## password ##
    #
    password:

      ## hooks ##
      #
      # Set this value using environment variables on
      # - Linux/macOS:
      #    $ export SELFSERVICE_FLOWS_REGISTRATION_AFTER_PASSWORD_HOOKS=<value>
      # - Windows Command Line (CMD):
      #    > set SELFSERVICE_FLOWS_REGISTRATION_AFTER_PASSWORD_HOOKS=<value>
      #
      hooks:
        - hook: session
        - hook: show_verification_ui

  ## Enable User Registration ##
  #
  # If set to true will enable [User Registration](https://www.ory.sh/kratos/docs/self-s
  #
  # Default value: true
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SELFSERVICE_FLOWS_REGISTRATION_ENABLED=<value>
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_FLOWS_REGISTRATION_ENABLED=<value>
  #
  enabled: true

## login ##
#
login:

  ## lifespan ##
  #
  # Default value: 1h
  #
  # Examples:
  # - 1h
  # - 1m
  # - 1s
  #
```

```
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SELFSERVICE_FLOWS_LOGIN_LIFESPAN=<value>
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_FLOWS_LOGIN_LIFESPAN=<value>
  #
  lifespan: 1h

  ## Login UI URL ##
  #
  # URL where the Login UI is hosted. Check the [reference implementation](https://githu
  #
  # Default value: https://www.ory.sh/kratos/docs/fallback/login
  #
  # Examples:
  # - https://my-app.com/login
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SELFSERVICE_FLOWS_LOGIN_UI_URL=<value>
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_FLOWS_LOGIN_UI_URL=<value>
  #
  ui_url: https://auth.example.com/login

## Email and Phone Verification and Account Activation Configuration ##
#
verification:

  ## Verify UI URL ##
  #
  # URL where the Ory Verify UI is hosted. This is the page where users activate and / o
  #
  # Default value: https://www.ory.sh/kratos/docs/fallback/verification
  #
  # Examples:
  # - https://my-app.com/verify
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SELFSERVICE_FLOWS_VERIFICATION_UI_URL=<value>
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_FLOWS_VERIFICATION_UI_URL=<value>
  #
  ui_url: https://auth.example.com/verification
```

```yaml
## after ##
#
after:

  ## Redirect browsers to set URL per default ##
  #
  # Ory Kratos redirects to this URL per default on completion of self-service flows
  #
  # Examples:
  # - https://my-app.com/dashboard
  # - /dashboard
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SELFSERVICE_FLOWS_VERIFICATION_AFTER_DEFAULT_BROWSER_RETURN_URL=<value
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_FLOWS_VERIFICATION_AFTER_DEFAULT_BROWSER_RETURN_URL=<value>
  #
  default_browser_return_url: https://auth.example.com/

## Self-Service Verification Request Lifespan ##
#
# Sets how long the verification request (for the UI interaction) is valid.
#
# Default value: 1h
#
# Examples:
# - 1h
# - 1m
# - 1s
#
# Set this value using environment variables on
# - Linux/macOS:
#    $ export SELFSERVICE_FLOWS_VERIFICATION_LIFESPAN=<value>
# - Windows Command Line (CMD):
#    > set SELFSERVICE_FLOWS_VERIFICATION_LIFESPAN=<value>
#
lifespan: 1h

## Verification Strategy ##
#
# The strategy to use for verification requests
#
# Default value: code
#
# One of:
```

```
  # - link
  # - code
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SELFSERVICE_FLOWS_VERIFICATION_USE=<value>
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_FLOWS_VERIFICATION_USE=<value>
  #
  use: code

  ## Enable Email/Phone Verification ##
  #
  # If set to true will enable [Email and Phone Verification and Account Activation](ht
  #
  # Default value: false
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SELFSERVICE_FLOWS_VERIFICATION_ENABLED=<value>
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_FLOWS_VERIFICATION_ENABLED=<value>
  #
  enabled: true

## Account Recovery Configuration ##
#
recovery:

  ## Recovery UI URL ##
  #
  # URL where the Ory Recovery UI is hosted. This is the page where users request and c
  #
  # Default value: https://www.ory.sh/kratos/docs/fallback/recovery
  #
  # Examples:
  # - https://my-app.com/verify
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SELFSERVICE_FLOWS_RECOVERY_UI_URL=<value>
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_FLOWS_RECOVERY_UI_URL=<value>
  #
  ui_url: https://auth.example.com/recovery
```

```yaml
## Self-Service Recovery Request Lifespan ##
#
# Sets how long the recovery request is valid. If expired, the user has to redo the f
#
# Default value: 1h
#
# Examples:
# - 1h
# - 1m
# - 1s
#
# Set this value using environment variables on
# - Linux/macOS:
#    $ export SELFSERVICE_FLOWS_RECOVERY_LIFESPAN=<value>
# - Windows Command Line (CMD):
#    > set SELFSERVICE_FLOWS_RECOVERY_LIFESPAN=<value>
#
lifespan: 1h

## Recovery Strategy ##
#
# The strategy to use for recovery requests
#
# Default value: code
#
# One of:
# - link
# - code
#
# Set this value using environment variables on
# - Linux/macOS:
#    $ export SELFSERVICE_FLOWS_RECOVERY_USE=<value>
# - Windows Command Line (CMD):
#    > set SELFSERVICE_FLOWS_RECOVERY_USE=<value>
#
use: code

## Enable Account Recovery ##
#
# If set to true will enable [Account Recovery](https://www.ory.sh/kratos/docs/self-s
#
# Default value: false
#
# Set this value using environment variables on
# - Linux/macOS:
#    $ export SELFSERVICE_FLOWS_RECOVERY_ENABLED=<value>
```

```
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_FLOWS_RECOVERY_ENABLED=<value>
  #
  enabled: true

## error ##
#
error:

  ## Ory Kratos Error UI URL ##
  #
  # URL where the Ory Kratos Error UI is hosted. Check the [reference implementation](h
  #
  # Default value: https://www.ory.sh/kratos/docs/fallback/error
  #
  # Examples:
  # - https://my-app.com/kratos-error
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SELFSERVICE_FLOWS_ERROR_UI_URL=<value>
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_FLOWS_ERROR_UI_URL=<value>
  #
  ui_url: https://auth.example.com/error

## settings ##
#
settings:

  ## lifespan ##
  #
  # Default value: 1h
  #
  # Examples:
  # - 1h
  # - 1m
  # - 1s
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SELFSERVICE_FLOWS_SETTINGS_LIFESPAN=<value>
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_FLOWS_SETTINGS_LIFESPAN=<value>
  #
  lifespan: 1h
```

```yaml
## privileged_session_max_age ##
#
# Default value: 1h
#
# Examples:
# - 1h
# - 1m
# - 1s
#
# Set this value using environment variables on
# - Linux/macOS:
#    $ export SELFSERVICE_FLOWS_SETTINGS_PRIVILEGED_SESSION_MAX_AGE=<value>
# - Windows Command Line (CMD):
#    > set SELFSERVICE_FLOWS_SETTINGS_PRIVILEGED_SESSION_MAX_AGE=<value>
#
privileged_session_max_age: 1h

## Required Authenticator Assurance Level ##
#
# Sets what Authenticator Assurance Level (used for 2FA) is required to access this f
#
# Default value: highest_available
#
# One of:
# - aal1
# - highest_available
#
# Set this value using environment variables on
# - Linux/macOS:
#    $ export SELFSERVICE_FLOWS_SETTINGS_REQUIRED_AAL=<value>
# - Windows Command Line (CMD):
#    > set SELFSERVICE_FLOWS_SETTINGS_REQUIRED_AAL=<value>
#
required_aal: highest_available

## URL of the Settings page. ##
#
# URL where the Settings UI is hosted. Check the [reference implementation](https://g
#
# Default value: https://www.ory.sh/kratos/docs/fallback/settings
#
# Examples:
# - https://my-app.com/user/settings
#
# Set this value using environment variables on
```

```
    # - Linux/macOS:
    #    $ export SELFSERVICE_FLOWS_SETTINGS_UI_URL=<value>
    # - Windows Command Line (CMD):
    #    > set SELFSERVICE_FLOWS_SETTINGS_UI_URL=<value>
    #
    ui_url: https://auth.example.com/settings

## methods ##
#
methods:

  ## link ##
  #
  link:

    ## Link Configuration ##
    #
    # Additional configuration for the link strategy.
    #
    config:

      ## How long a link is valid for ##
      #
      # Default value: 1h
      #
      # Examples:
      # - 1h
      # - 1m
      # - 1s
      #
      # Set this value using environment variables on
      # - Linux/macOS:
      #    $ export SELFSERVICE_METHODS_LINK_CONFIG_LIFESPAN=<value>
      # - Windows Command Line (CMD):
      #    > set SELFSERVICE_METHODS_LINK_CONFIG_LIFESPAN=<value>
      #
      lifespan: 1h

    ## Enables Link Method ##
    #
    # Default value: false
    #
    # Set this value using environment variables on
    # - Linux/macOS:
    #    $ export SELFSERVICE_METHODS_LINK_ENABLED=<value>
    # - Windows Command Line (CMD):
```

```
  #    > set SELFSERVICE_METHODS_LINK_ENABLED=<value>
  #
  enabled: true

## code ##
#
code:

  ## Enables Code Method ##
  #
  # Default value: true
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SELFSERVICE_METHODS_CODE_ENABLED=<value>
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_METHODS_CODE_ENABLED=<value>
  #
  enabled: true

  ## Code Configuration ##
  #
  # Additional configuration for the code strategy.
  #
  config:

    ## How long a code is valid for ##
    #
    # Default value: 1h
    #
    # Examples:
    # - 1h
    # - 1m
    # - 1s
    #
    # Set this value using environment variables on
    # - Linux/macOS:
    #    $ export SELFSERVICE_METHODS_CODE_CONFIG_LIFESPAN=<value>
    # - Windows Command Line (CMD):
    #    > set SELFSERVICE_METHODS_CODE_CONFIG_LIFESPAN=<value>
    #
    lifespan: 1h

## password ##
#
password:
```

```yaml
## Password Configuration ##
#
# Define how passwords are validated.
#
config:

  ## Enable the HaveIBeenPwned API ##
  #
  # If set to false the password validation does not utilize the Have I Been Pwnd API
  #
  # Default value: true
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SELFSERVICE_METHODS_PASSWORD_CONFIG_HAVEIBEENPWNED_ENABLED=<value>
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_METHODS_PASSWORD_CONFIG_HAVEIBEENPWNED_ENABLED=<value>
  #
  haveibeenpwned_enabled: false

  ## Allow Password Breaches ##
  #
  # Defines how often a password may have been breached before it is rejected.
  #
  # Minimum value: 0
  #
  # Maximum value: 100
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SELFSERVICE_METHODS_PASSWORD_CONFIG_MAX_BREACHES=<value>
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_METHODS_PASSWORD_CONFIG_MAX_BREACHES=<value>
  #
  max_breaches: 0

  ## Minimum Password Length ##
  #
  # Defines the minimum length of the password.
  #
  # Default value: 8
  #
  # Minimum value: 6
  #
  # Set this value using environment variables on
```

```
    # - Linux/macOS:
    #    $ export SELFSERVICE_METHODS_PASSWORD_CONFIG_MIN_PASSWORD_LENGTH=<value>
    # - Windows Command Line (CMD):
    #    > set SELFSERVICE_METHODS_PASSWORD_CONFIG_MIN_PASSWORD_LENGTH=<value>
    #
    min_password_length: 6

    ## Enable password-identifier similarity check ##
    #
    # If set to false the password validation does not check for similarity between the
    #
    # Default value: true
    #
    # Set this value using environment variables on
    # - Linux/macOS:
    #    $ export SELFSERVICE_METHODS_PASSWORD_CONFIG_IDENTIFIER_SIMILARITY_CHECK_ENABL
    # - Windows Command Line (CMD):
    #    > set SELFSERVICE_METHODS_PASSWORD_CONFIG_IDENTIFIER_SIMILARITY_CHECK_ENABLED=
    #
    identifier_similarity_check_enabled: true

  ## Enables Username/Email and Password Method ##
  #
  # Default value: true
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SELFSERVICE_METHODS_PASSWORD_ENABLED=<value>
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_METHODS_PASSWORD_ENABLED=<value>
  #
  enabled: true

## totp ##
#
totp:

  ## TOTP Configuration ##
  #
  config:

    ## TOTP Issuer ##
    #
    # The issuer (e.g. a domain name) will be shown in the TOTP app (e.g. Google Authen
    #
    # Set this value using environment variables on
```

```
      # - Linux/macOS:
      #    $ export SELFSERVICE_METHODS_TOTP_CONFIG_ISSUER=<value>
      # - Windows Command Line (CMD):
      #    > set SELFSERVICE_METHODS_TOTP_CONFIG_ISSUER=<value>
      #
      issuer: Kratos

    ## Enables the TOTP method ##
    #
    # Default value: false
    #
    # Set this value using environment variables on
    # - Linux/macOS:
    #    $ export SELFSERVICE_METHODS_TOTP_ENABLED=<value>
    # - Windows Command Line (CMD):
    #    > set SELFSERVICE_METHODS_TOTP_ENABLED=<value>
    #
    enabled: true

  ## Allowed Return To URLs ##
  #
  # List of URLs that are allowed to be redirected to. A redirection request is made by appe
  #
  # Examples:
  # - - https://app.my-app.com/dashboard
  #   - /dashboard
  #   - https://www.my-app.com/
  #   - https://*.my-app.com/
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SELFSERVICE_ALLOWED_RETURN_URLS=<value>
  # - Windows Command Line (CMD):
  #    > set SELFSERVICE_ALLOWED_RETURN_URLS=<value>
  #
  allowed_return_urls:
    - https://*.example.com

## Courier configuration ##
#
# The courier is responsible for sending and delivering messages over email, sms, and other
#
courier:

  ## SMTP Configuration ##
  #
```

```yaml
# Configures outgoing emails using the SMTP protocol.
#
smtp:

  ## SMTP connection string ##
  #
  # This URI will be used to connect to the SMTP server. Use the scheme smtps for implici
  #
  # Examples:
  # - smtps://foo:bar@my-mailserver:1234/?skip_ssl_verify=false
  # - "smtp://foo:bar@my-mailserver:1234/?disable_starttls=true (NOT RECOMMENDED:
  #   Cleartext smtp for devel and legacy infrastructure only)"
  # - smtp://foo:bar@my-mailserver:1234/ (Explicit StartTLS with certificate trust
  #   verification)
  # - "smtp://foo:bar@my-mailserver:1234/?skip_ssl_verify=true (NOT RECOMMENDED:
  #   Explicit StartTLS without certificate trust verification)"
  # - smtps://foo:bar@my-mailserver:1234/ (Implicit TLS with certificate trust
  #   verification)
  # - "smtps://foo:bar@my-mailserver:1234/?skip_ssl_verify=true (NOT RECOMMENDED:
  #   Implicit TLS without certificate trust verification)"
  # - smtps://subdomain.my-mailserver:1234/?server_name=my-mailserver (allows TLS to
  #   work if the server is hosted on a sudomain that uses a non-wildcard domain
  #   certificate)
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export COURIER_SMTP_CONNECTION_URI=<value>
  # - Windows Command Line (CMD):
  #    > set COURIER_SMTP_CONNECTION_URI=<value>
  #
  connection_uri: smtps://foo:bar@my-mailserver:1234

  ## SMTP Sender Address ##
  #
  # The recipient of an email will see this as the sender address.
  #
  # Default value: no-reply@ory.kratos.sh
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export COURIER_SMTP_FROM_ADDRESS=<value>
  # - Windows Command Line (CMD):
  #    > set COURIER_SMTP_FROM_ADDRESS=<value>
  #
  from_address: tuan.nguyen930708@gmail.com
```

```
  ## SMTP Sender Name ##
  #
  # The recipient of an email will see this as the sender name.
  #
  # Examples:
  # - Bob
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export COURIER_SMTP_FROM_NAME=<value>
  # - Windows Command Line (CMD):
  #    > set COURIER_SMTP_FROM_NAME=<value>
  #
  from_name: "Tuan Nguyen"

  ## SMTP Headers ##
  #
  # These headers will be passed in the SMTP conversation -- e.g. when using the AWS SES S
  #
  # Examples:
  # - X-SES-SOURCE-ARN: arn:aws:ses:us-west-2:123456789012:identity/example.com
  #   X-SES-FROM-ARN: arn:aws:ses:us-west-2:123456789012:identity/example.com
  #   X-SES-RETURN-PATH-ARN: arn:aws:ses:us-west-2:123456789012:identity/example.com
  #
  headers:

    ## X-SES-FROM-ARN ##
    #
    # Set this value using environment variables on
    # - Linux/macOS:
    #    $ export COURIER_SMTP_HEADERS_X-SES-FROM-ARN=<value>
    # - Windows Command Line (CMD):
    #    > set COURIER_SMTP_HEADERS_X-SES-FROM-ARN=<value>
    #
    X-SES-FROM-ARN: arn:aws:ses:us-west-2:123456789012:identity/example.com

## message_retries ##
#
# Defines the maximum number of times the sending of a message is retried after it failed
#
# Default value: 5
#
# Examples:
# - 10
# - 60
#
```

```
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export COURIER_MESSAGE_RETRIES=<value>
  # - Windows Command Line (CMD):
  #    > set COURIER_MESSAGE_RETRIES=<value>
  #
  message_retries: 5

  ## Delivery Strategy ##
  #
  # Defines how emails will be sent, either through SMTP (default) or HTTP.
  #
  # Default value: smtp
  #
  # One of:
  # - smtp
  # - http
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export COURIER_DELIVERY_STRATEGY=<value>
  # - Windows Command Line (CMD):
  #    > set COURIER_DELIVERY_STRATEGY=<value>
  #
  delivery_strategy: smtp

## serve ##
#
serve:

  ## public ##
  #
  public:

    ## cors ##
    #
    # Configures Cross Origin Resource Sharing for public endpoints.
    #
    cors:

      ## allowed_origins ##
      #
      # A list of origins a cross-domain request can be executed from. If the special * valu
      #
      # Default value: *
      #
```

```yaml
# Examples:
# - - https://example.com
#   - https://*.example.com
#   - https://*.foo.example.com
#
# Set this value using environment variables on
# - Linux/macOS:
#    $ export SERVE_PUBLIC_CORS_ALLOWED_ORIGINS=<value>
# - Windows Command Line (CMD):
#    > set SERVE_PUBLIC_CORS_ALLOWED_ORIGINS=<value>
#
allowed_origins:
  - https://example.com
  - https://*.example.com

## allowed_methods ##
#
# A list of HTTP methods the user agent is allowed to use with cross-domain requests.
#
# Default value: POST,GET,PUT,PATCH,DELETE
#
# Set this value using environment variables on
# - Linux/macOS:
#    $ export SERVE_PUBLIC_CORS_ALLOWED_METHODS=<value>
# - Windows Command Line (CMD):
#    > set SERVE_PUBLIC_CORS_ALLOWED_METHODS=<value>
#
allowed_methods:
  - POST
  - GET
  - PUT
  - PATCH
  - DELETE

## allowed_headers ##
#
# A list of non simple headers the client is allowed to use with cross-domain requests
#
# Default value: Authorization,Content-Type,Max-Age,X-Session-Token,X-XSRF-TOKEN,X-CSR
#
# Set this value using environment variables on
# - Linux/macOS:
#    $ export SERVE_PUBLIC_CORS_ALLOWED_HEADERS=<value>
# - Windows Command Line (CMD):
#    > set SERVE_PUBLIC_CORS_ALLOWED_HEADERS=<value>
#
```

```yaml
allowed_headers:
  - Authorization
  - Content-Type
  - Max-Age
  - X-Session-Token
  - X-XSRF-TOKEN
  - X-CSRF-TOKEN

## exposed_headers ##
#
# Sets which headers are safe to expose to the API of a CORS API specification.
#
# Default value: Content-Type
#
# Set this value using environment variables on
# - Linux/macOS:
#    $ export SERVE_PUBLIC_CORS_EXPOSED_HEADERS=<value>
# - Windows Command Line (CMD):
#    > set SERVE_PUBLIC_CORS_EXPOSED_HEADERS=<value>
#
exposed_headers:
  - Content-Type

## allow_credentials ##
#
# Sets whether the request can include user credentials like cookies, HTTP authentica
#
# Default value: true
#
# Set this value using environment variables on
# - Linux/macOS:
#    $ export SERVE_PUBLIC_CORS_ALLOW_CREDENTIALS=<value>
# - Windows Command Line (CMD):
#    > set SERVE_PUBLIC_CORS_ALLOW_CREDENTIALS=<value>
#
allow_credentials: true

## max_age ##
#
# Sets how long (in seconds) the results of a preflight request can be cached. If set
#
# Minimum value: 0
#
# Set this value using environment variables on
# - Linux/macOS:
#    $ export SERVE_PUBLIC_CORS_MAX_AGE=<value>
```

```
    # - Windows Command Line (CMD):
    #    > set SERVE_PUBLIC_CORS_MAX_AGE=<value>
    #
    max_age: 3600

    ## enabled ##
    #
    # Sets whether CORS is enabled.
    #
    # Default value: false
    #
    # Set this value using environment variables on
    # - Linux/macOS:
    #    $ export SERVE_PUBLIC_CORS_ENABLED=<value>
    # - Windows Command Line (CMD):
    #    > set SERVE_PUBLIC_CORS_ENABLED=<value>
    #
    enabled: true

  ## Base URL ##
  #
  # The URL where the endpoint is exposed at. This domain is used to generate redirects, f
  #
  # Examples:
  # - https://my-app.com/
  # - https://my-app.com/.ory/kratos/public
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SERVE_PUBLIC_BASE_URL=<value>
  # - Windows Command Line (CMD):
  #    > set SERVE_PUBLIC_BASE_URL=<value>
  #
  base_url: https://kratos.example.com/

## admin ##
#
admin:

  ## Admin Base URL ##
  #
  # The URL where the admin endpoint is exposed at.
  #
  # Examples:
  # - https://kratos.private-network:4434/
  #
```

```
    # Set this value using environment variables on
    # - Linux/macOS:
    #    $ export SERVE_ADMIN_BASE_URL=<value>
    # - Windows Command Line (CMD):
    #    > set SERVE_ADMIN_BASE_URL=<value>
    #
    base_url: http://kratos.svc.cluster.local:4434/

## Log ##
#
# Configure logging using the following options. Logging will always be sent to stdout and s
#
log:

  ## Leak Sensitive Log Values ##
  #
  # If set will leak sensitive values (e.g. emails) in the logs.
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export LOG_LEAK_SENSITIVE_VALUES=<value>
  # - Windows Command Line (CMD):
  #    > set LOG_LEAK_SENSITIVE_VALUES=<value>
  #
  leak_sensitive_values: false

  ## Sensitive log value redaction text ##
  #
  # Text to use, when redacting sensitive log value.
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export LOG_REDACTION_TEXT=<value>
  # - Windows Command Line (CMD):
  #    > set LOG_REDACTION_TEXT=<value>
  #
  redaction_text: ""

  ## format ##
  #
  # The log format can either be text or JSON.
  #
  # One of:
  # - json
  # - text
  #
```

```yaml
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export LOG_FORMAT=<value>
  # - Windows Command Line (CMD):
  #    > set LOG_FORMAT=<value>
  #
  format: json

  ## level ##
  #
  # Debug enables stack traces on errors. Can also be set using environment variable LOG_LE
  #
  # Default value: info
  #
  # One of:
  # - trace
  # - debug
  # - info
  # - warning
  # - error
  # - fatal
  # - panic
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export LOG_LEVEL=<value>
  # - Windows Command Line (CMD):
  #    > set LOG_LEVEL=<value>
  #
  level: trace

## secrets ##
#
secrets:

  ## Signing Keys for Cookies ##
  #
  # The first secret in the array is used for encrypting cookies while all other keys are us
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SECRETS_COOKIE=<value>
  # - Windows Command Line (CMD):
  #    > set SECRETS_COOKIE=<value>
  #
  cookie:
```

```
    - a8ba727309b345a0a0e8dd9b2eae8b19

  ## Secrets to use for encryption by cipher ##
  #
  # The first secret in the array is used for encryption data while all other keys are used
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SECRETS_CIPHER=<value>
  # - Windows Command Line (CMD):
  #    > set SECRETS_CIPHER=<value>
  #
  cipher:
    - feaf396bf5ca4c93870782f0df4b7b03

## Hashing Algorithm Configuration ##
#
hashers:
  ## Configuration for the Bcrypt hasher. Minimum is 4 when --dev flag is used and 12 otherw
  #
  bcrypt:

    ## cost ##
    #
    # Default value: 12
    #
    # Minimum value: 4
    #
    # Maximum value: 31
    #
    # Set this value using environment variables on
    # - Linux/macOS:
    #    $ export HASHERS_BCRYPT_COST=<value>
    # - Windows Command Line (CMD):
    #    > set HASHERS_BCRYPT_COST=<value>
    #
    cost: 12

  ## Password hashing algorithm ##
  #
  # One of the values: argon2, bcrypt.
  # Any other hashes will be migrated to the set algorithm once an identity authenticates us
  #
  # Default value: bcrypt
  #
  # One of:
```

```
  # - argon2
  # - bcrypt
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export HASHERS_ALGORITHM=<value>
  # - Windows Command Line (CMD):
  #    > set HASHERS_ALGORITHM=<value>
  #
  algorithm: bcrypt

## Cipher Algorithm Configuration ##
#
ciphers:

  ## ciphering algorithm ##
  #
  # One of the values: noop, aes, xchacha20-poly1305
  #
  # Default value: noop
  #
  # One of:
  # - noop
  # - aes
  # - xchacha20-poly1305
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export CIPHERS_ALGORITHM=<value>
  # - Windows Command Line (CMD):
  #    > set CIPHERS_ALGORITHM=<value>
  #
  algorithm: xchacha20-poly1305

## HTTP Cookie Configuration ##
#
# Configure the HTTP Cookies. Applies to both CSRF and session cookies.
#
cookies:
  ## HTTP Cookie Domain ##
  #
  # Sets the cookie domain for session and CSRF cookies. Useful when dealing with subdomains
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export COOKIES_DOMAIN=<value>
```

```
  # - Windows Command Line (CMD):
  #    > set COOKIES_DOMAIN=<value>
  #
  domain: ".example.com"

## session ##
#
session:

  ## Session Lifespan ##
  #
  # Defines how long a session is active. Once that lifespan has been reached, the user nee
  #
  # Default value: 24h
  #
  # Examples:
  # - 1h
  # - 1m
  # - 1s
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SESSION_LIFESPAN=<value>
  # - Windows Command Line (CMD):
  #    > set SESSION_LIFESPAN=<value>
  #
  lifespan: 24h

  ## Earliest Possible Session Extension ##
  #
  # Sets when a session can be extended. Settings this value to `24h` will prevent the sess
  #
  # Examples:
  # - 1h
  # - 1m
  # - 1s
  #
  # Set this value using environment variables on
  # - Linux/macOS:
  #    $ export SESSION_EARLIEST_POSSIBLE_EXTEND=<value>
  # - Windows Command Line (CMD):
  #    > set SESSION_EARLIEST_POSSIBLE_EXTEND=<value>
  #
  earliest_possible_extend: 1h

  ## WhoAmI / ToSession Settings ##
```

```yaml
  #
  # Control how the `/sessions/whoami` endpoint is behaving.
  #
  whoami:
    ## Required Authenticator Assurance Level ##
    #
    # Sets what Authenticator Assurance Level (used for 2FA) is required to access this fea
    #
    # Default value: highest_available
    #
    # One of:
    # - aal1
    # - highest_available
    #
    # Set this value using environment variables on
    # - Linux/macOS:
    #    $ export SESSION_WHOAMI_REQUIRED_AAL=<value>
    # - Windows Command Line (CMD):
    #    > set SESSION_WHOAMI_REQUIRED_AAL=<value>
    #
    required_aal: aal1

## The kratos version this config is written for. ##
#
# SemVer according to https://semver.org/ prefixed with `v` as in our releases.
#
# Examples:
# - v0.5.0-alpha.1
#
# Set this value using environment variables on
# - Linux/macOS:
#    $ export VERSION=<value>
# - Windows Command Line (CMD):
#    > set VERSION=<value>
#
version: v1.0.0

## dev ##
#
# Set this value using environment variables on
# - Linux/macOS:
#    $ export DEV=<value>
# - Windows Command Line (CMD):
#    > set DEV=<value>
#
dev: false
```

```
## help ##
#
# Set this value using environment variables on
# - Linux/macOS:
#    $ export HELP=<value>
# - Windows Command Line (CMD):
#    > set HELP=<value>
#
help: false
```

Kubernetes Secret

```
apiVersion: v1kind: Secret
metadata:
  name: identity-configs
  namespace: identity
data:
  configs.yaml: dmVyc2lvbjogdjEuMC4wCgpkc246IHBvc3RncmVzOi8va2FudGhvcjozMTI1Njg2Mi05ZDQ1LTQ1
```

Deployment

Make sure you replace `scrapnode/kratos:uat-f195a5d6-202312191451` with your docker image of Kratos

Kubernetes Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: kratos
  namespace: identity
  labels:
    app.kubernetes.io/name: kratos
    app.kubernetes.io/version: "v1.0.0"
spec:
  replicas: 5
  selector:
    matchLabels:
      app.kubernetes.io/name: kratos
      app.kubernetes.io/version: "v1.0.0"
  template:
    metadata:
      labels:
        app.kubernetes.io/name: kratos
        app.kubernetes.io/version: "v1.0.0"
    spec:
      volumes:
```

```
  - name: schemas
    configMap:
      name: identity-schemas
  - name: configs
    secret:
      secretName: identity-configs
initContainers:
  - name: migrate
    image: scrapnode/kratos:uat-f195a5d6-202312191451
    volumeMounts:
      - name: schemas
        mountPath: /identity/schemas
      - name: configs
        mountPath: /identity/configs
    args:
      - -c
      - /identity/configs/configs.yaml
      - migrate
      - sql
      - -e
      - --yes
containers:
  - name: kratos
    image: scrapnode/kratos:uat-f195a5d6-202312191451
    args:
      - -c
      - /identity/configs/configs.yaml
      - serve
    volumeMounts:
      - name: schemas
        mountPath: /identity/schemas
      - name: configs
        mountPath: /identity/configs
    ports:
      - name: public
        containerPort: 4433
        protocol: TCP
      - name: admin
        containerPort: 4434
        protocol: TCP
  - name: courier
    image: oryd/kratos:v1.0.0
    args:
      - -c
      - /identity/configs/configs.yaml
      - courier
```

```
            - watch
          volumeMounts:
            - name: schemas
              mountPath: /identity/schemas
            - name: configs
              mountPath: /identity/configs
---
apiVersion: v1
kind: Service
metadata:
  name: kratos-public
  namespace: identity
  labels:
    app.kubernetes.io/name: kratos
    app.kubernetes.io/version: "v1.0.0"
spec:
  selector:
    app.kubernetes.io/name: kratos
    app.kubernetes.io/version: "v1.0.0"
  ports:
    - port: 4433
      targetPort: public
      protocol: TCP
      name: public
---
apiVersion: v1
kind: Service
metadata:
  name: kratos-admin
  namespace: identity
  labels:
    app.kubernetes.io/name: kratos
    app.kubernetes.io/version: "v1.0.0"
spec:
  selector:
    app.kubernetes.io/name: kratos
    app.kubernetes.io/version: "v1.0.0"
  ports:
    - port: 4434
      targetPort: admin
      protocol: TCP
      name: admin
```

Integration

Administrator

In this guideline I am going to use the domain `http://kratos.svc.cluster.local:4434` as the host of admin API, you NEED TO change it after configure the deployment. The host must be setup with SSL certificate to secure your API

The Administrator APIs don't come with integrated access control. This means that all requests sent to their APIs are considered authenticated, authorized, and will be executed. So you should only expose those APIs to trust servers or netwokrs. The best practise is using the Administrator APIs inside a private network with server-server comminucation

Create new account Options:

- `traits.email`: the email of new account
- `credentials.password.config.password` the password of new account

Request:

```
curl --location 'http://kratos.svc.cluster.local:4434/admin/identities' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'Authorization;' \
--data-raw '{
    "schema_id": "default",
    "traits": {
        "email": "tuan.nguyen930708@gmail.com"
    },
    "credentials": {
        "password": {
            "config": {
                "password": "thepassword_thatis_STRONG"
            }
        }
    }
}
'
```

Response Body:

```
{
    "id": "7ca3df0e-874b-44f7-a258-6223d49f5952",
    "credentials": {
        "password": {
            "type": "password",
            "identifiers": [
```

```
                "tuan.nguyen930708@gmail.com"
            ],
            "version": 0,
            "created_at": "0001-01-01T00:00:00Z",
            "updated_at": "0001-01-01T00:00:00Z"
        }
    },
    "schema_id": "default",
    "schema_url": "https://kratos.example.com/schemas/ZGVmYXVsdA",
    "state": "active",
    "state_changed_at": "2023-12-20T07:20:43.511500574Z",
    "traits": {
        "email": "tuan.nguyen930708@gmail.com"
    },
    "verifiable_addresses": [
        {
            "id": "688c6fc6-0d75-49bb-93ed-0860d1c600c0",
            "value": "tuan.nguyen930708@gmail.com",
            "verified": false,
            "via": "email",
            "status": "pending",
            "created_at": "2023-12-20T07:20:43.513698Z",
            "updated_at": "2023-12-20T07:20:43.513698Z"
        }
    ],
    "recovery_addresses": [
        {
            "id": "c71a5f75-7865-47ce-93c4-28bd3248fada",
            "value": "tuan.nguyen930708@gmail.com",
            "via": "email",
            "created_at": "2023-12-20T07:20:43.514484Z",
            "updated_at": "2023-12-20T07:20:43.514484Z"
        }
    ],
    "metadata_public": null,
    "created_at": "2023-12-20T07:20:43.512549Z",
    "updated_at": "2023-12-20T07:20:43.512549Z",
    "organization_id": null
}
```


Get an account Options

- include_credential
  - `password`: return the password credentials (with the `hashed_password` attributes)
  - `oidc`: return the initial OAuth 2.0 Access Token, OAuth 2.0 Refresh

Token and the OpenID Connect ID Token if available.

Request

```
curl --location http://kratos.svc.cluster.local:4434/admin/identities/7ca3df0e-874b-44f7-a25
--header 'Accept: application/json'
```

Response Body


```
{
    "id": "7ca3df0e-874b-44f7-a258-6223d49f5952",
    "credentials": {
        "password": {
            "type": "password",
            "identifiers": [
                "tuan.nguyen930708@gmail.com"
            ],
            "config": {
                "hashed_password": "$2a$12$/dkPt64Ngazx/acXfp8tTOwCRYGNThCm9rZNyrt1a2Ygtjksa
            },
            "version": 0,
            "created_at": "2023-12-20T07:20:43.515658Z",
            "updated_at": "2023-12-20T07:20:43.515658Z"
        }
    },
    "schema_id": "default",
    "schema_url": "https://kratos.example.com/schemas/ZGVmYXVsdA",
    "state": "active",
    "state_changed_at": "2023-12-20T07:20:43.5115Z",
    "traits": {
        "email": "tuan.nguyen930708@gmail.com"
    },
    "verifiable_addresses": [
        {
            "id": "688c6fc6-0d75-49bb-93ed-0860d1c600c0",
            "value": "tuan.nguyen930708@gmail.com",
            "verified": false,
            "via": "email",
            "status": "pending",
            "created_at": "2023-12-20T07:20:43.513698Z",
            "updated_at": "2023-12-20T07:20:43.513698Z"
        }
    ],
    "recovery_addresses": [
        {
            "id": "c71a5f75-7865-47ce-93c4-28bd3248fada",
            "value": "tuan.nguyen930708@gmail.com",
            "via": "email",
```

```
            "created_at": "2023-12-20T07:20:43.514484Z",
            "updated_at": "2023-12-20T07:20:43.514484Z"
        }
    ],
    "metadata_public": null,
    "metadata_admin": null,
    "created_at": "2023-12-20T07:20:43.512549Z",
    "updated_at": "2023-12-20T07:20:43.512549Z",
    "organization_id": null
}
```

Update an account There are various values you can add or replace, there are some usecase that may useful for you

Mark an email as verified manually Request

```
curl --location --request PATCH http://kratos.svc.cluster.local:4434/admin/identities/7ca3df
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--data '[
    {
        "op": "replace",
        "path": "/verifiable_addresses/0/verified",
        "value": true
    },
    {
        "op": "replace",
        "path": "/verifiable_addresses/0/status",
        "value": "completed"
    }
]'
```

Response Body

```
{
    "id": "7ca3df0e-874b-44f7-a258-6223d49f5952",
    "credentials": {
        "password": {
            "type": "password",
            "identifiers": [
                "tuan.nguyen930708@gmail.com"
            ],
            "version": 0,
            "created_at": "2023-12-20T07:20:43.515658Z",
            "updated_at": "2023-12-20T07:20:43.515658Z"
        }
    },
```

```
    "schema_id": "default",
    "schema_url": "https://kratos.example.com/schemas/ZGVmYXVsdA",
    "state": "active",
    "state_changed_at": "2023-12-20T07:20:43.5115Z",
    "traits": {
        "email": "tuan.nguyen930708@gmail.com"
    },
    "verifiable_addresses": [
        {
            "id": "688c6fc6-0d75-49bb-93ed-0860d1c600c0",
            "value": "tuan.nguyen930708@gmail.com",
            "verified": true,
            "via": "email",
            "status": "completed",
            "verified_at": "2023-12-20T07:37:51.488354267Z",
            "created_at": "2023-12-20T07:20:43.513698Z",
            "updated_at": "2023-12-20T07:37:51.490121Z"
        }
    ],
    "recovery_addresses": [
        {
            "id": "c71a5f75-7865-47ce-93c4-28bd3248fada",
            "value": "tuan.nguyen930708@gmail.com",
            "via": "email",
            "created_at": "2023-12-20T07:20:43.514484Z",
            "updated_at": "2023-12-20T07:20:43.514484Z"
        }
    ],
    "metadata_public": null,
    "metadata_admin": null,
    "created_at": "2023-12-20T07:20:43.512549Z",
    "updated_at": "2023-12-20T07:20:43.512549Z",
    "organization_id": null
}
```


Add some propeties to metadata_public (or metadata_admin) Request

```
curl --location --request PATCH http://kratos.svc.cluster.local:4434/admin/identities/7ca3df
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'Authorization;' \
--data '[
  {
    "op": "add",
    "path": "/metadata_public/everyone",
```

```
    "value": {
        "can": "read",
        "cannot": "modify"
    }
  },
  {
    "op": "add",
    "path": "/metadata_public/admin",
    "value": "can_do_whatever_admin_want_to"
  },
  {
    "op": "add",
    "path": "/metadata_admin/everyone",
    "value": {
        "cannot": "read_or_modify"
    }
  },
  {
    "op": "add",
    "path": "/metadata_admin/admin",
    "value": "can_do_whatever_admin_want_to"
  }
]'
```

Response Body


```
{
    "id": "7ca3df0e-874b-44f7-a258-6223d49f5952",
    "credentials": {
        "password": {
            "type": "password",
            "identifiers": [
                "tuan.nguyen930708@gmail.com"
            ],
            "version": 0,
            "created_at": "2023-12-20T07:20:43.515658Z",
            "updated_at": "2023-12-20T07:37:51.491144Z"
        }
    },
    "schema_id": "default",
    "schema_url": "https://kratos.example.com/schemas/ZGVmYXVsdA",
    "state": "active",
    "state_changed_at": "2023-12-20T07:20:43.5115Z",
    "traits": {
        "email": "tuan.nguyen930708@gmail.com"
    },
    "verifiable_addresses": [
```

```
        {
            "id": "688c6fc6-0d75-49bb-93ed-0860d1c600c0",
            "value": "tuan.nguyen930708@gmail.com",
            "verified": true,
            "via": "email",
            "status": "pending",
            "verified_at": "2023-12-20T07:37:51.488354Z",
            "created_at": "2023-12-20T07:20:43.513698Z",
            "updated_at": "2023-12-20T07:37:51.490121Z"
        }
    ],
    "recovery_addresses": [
        {
            "id": "c71a5f75-7865-47ce-93c4-28bd3248fada",
            "value": "tuan.nguyen930708@gmail.com",
            "via": "email",
            "created_at": "2023-12-20T07:20:43.514484Z",
            "updated_at": "2023-12-20T07:20:43.514484Z"
        }
    ],
    "metadata_public": {
        "everyone": {
            "can": "read",
            "cannot": "modify"
        },
        "admin": "can_do_whatever_admin_want_to"
    },
    "metadata_admin": {
        "everyone": {
            "cannot": "read_or_modify"
        },
        "admin": "can_do_whatever_admin_want_to"
    },
    "created_at": "2023-12-20T07:20:43.512549Z",
    "updated_at": "2023-12-20T07:20:43.512549Z",
    "organization_id": null
}
```


List accounts Options

- `per_page`: default is `250`
- `page`: default is `0`
- `credentials_identifier`: an email of the searching account if you want to find the account

Request

```
curl --location http://kratos.svc.cluster.local:4434/admin/identities?per_page=250&page=0&cr
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'Authorization;'
```

Response Body


```
[
    {
        "id": "7ca3df0e-874b-44f7-a258-6223d49f5952",
        "credentials": {
            "password": {
                "type": "password",
                "identifiers": [
                    "tuan.nguyen930708@gmail.com"
                ],
                "version": 0,
                "created_at": "2023-12-20T07:20:43.515658Z",
                "updated_at": "2023-12-20T07:50:39.846748Z"
            }
        },
        "schema_id": "default",
        "schema_url": "https://kratos.example.com/schemas/ZGVmYXVsdA",
        "state": "active",
        "state_changed_at": "2023-12-20T07:20:43.5115Z",
        "traits": {
            "email": "tuan.nguyen930708@gmail.com"
        },
        "verifiable_addresses": [
            {
                "id": "688c6fc6-0d75-49bb-93ed-0860d1c600c0",
                "value": "tuan.nguyen930708@gmail.com",
                "verified": true,
                "via": "email",
                "status": "completed",
                "verified_at": "2023-12-20T07:37:51.488354Z",
                "created_at": "2023-12-20T07:20:43.513698Z",
                "updated_at": "2023-12-20T07:50:39.845691Z"
            }
        ],
        "recovery_addresses": [
            {
                "id": "c71a5f75-7865-47ce-93c4-28bd3248fada",
                "value": "tuan.nguyen930708@gmail.com",
                "via": "email",
                "created_at": "2023-12-20T07:20:43.514484Z",
                "updated_at": "2023-12-20T07:20:43.514484Z"
```

```
            }
        ],
        "metadata_public": {
            "admin": "can_do_whatever_admin_want_to",
            "everyone": {
                "can": "read",
                "cannot": "modify"
            }
        },
        "metadata_admin": {
            "admin": "can_do_whatever_admin_want_to",
            "everyone": {
                "cannot": "read_or_modify"
            }
        },
        "created_at": "2023-12-20T07:20:43.512549Z",
        "updated_at": "2023-12-20T07:20:43.512549Z",
        "organization_id": null
    }
]
```


Delete an accounts Request

```
curl --location --request DELETE http://kratos.svc.cluster.local:4434/admin/identities/19fd3
--header 'Accept: application/json' \
--header 'Authorization;'
```

Frontend (Browser)

In this guideline I am going to use the domain `https://kratos.example.com` as the host of admin API, you NEED TO change it after configure the deployment. The host must be setup with SSL certificate to secure your API

IMPORANT NOTE: The browser API is heavily depended on cookkie-session model, so if you are wokring on Single Page Application, you need to send the `Cookie` header with each request you made to the Kratos Identity Server

Registration

1. Init the registration flow

   Request

   ```
   curl --location 'https://kratos.example.com/self-service/registration/browser' \
   --header 'Content-Type: application/json' \
   --header 'Accept: application/json' \
   ```

   Response Body

```
{
    "id": "eb2bfb29-7fe9-46ed-8785-539d990ed3ff",
    "type": "browser",
    "expires_at": "2023-12-20T09:51:01.895820353Z",
    "issued_at": "2023-12-20T09:41:01.895820353Z",
    "request_url": "https://kratos.example.com/self-service/registration/browser",
    "ui": {
        "action": "https://kratos.example.com/self-service/registration?flow=eb2bfb29-7
        "method": "POST",
        "nodes": [
            {
                "type": "input",
                "group": "default",
                "attributes": {
                    "name": "csrf_token",
                    "type": "hidden",
                    "value": "tmqMMK+4Ii9vIBzA9Xr5DzajAIaIXoyGKecGRbVQDde3cWugSjH+m2bXE
                    "required": true,
                    "disabled": false,
                    "node_type": "input"
                },
                "messages": [],
                "meta": {}
            },
            {
                "type": "input",
                "group": "password",
                "attributes": {
                    "name": "traits.email",
                    "type": "email",
                    "required": true,
                    "autocomplete": "email",
                    "disabled": false,
                    "node_type": "input"
                },
                "messages": [],
                "meta": {
                    "label": {
                        "id": 1070002,
                        "text": "E-Mail",
                        "type": "info",
                        "context": {
                            "title": "E-Mail"
                        }
                    }
                }
```

```
            },
            {
                "type": "input",
                "group": "password",
                "attributes": {
                    "name": "password",
                    "type": "password",
                    "required": true,
                    "autocomplete": "new-password",
                    "disabled": false,
                    "node_type": "input"
                },
                "messages": [],
                "meta": {
                    "label": {
                        "id": 1070001,
                        "text": "Password",
                        "type": "info"
                    }
                }
            },
            {
                "type": "input",
                "group": "password",
                "attributes": {
                    "name": "method",
                    "type": "submit",
                    "value": "password",
                    "disabled": false,
                    "node_type": "input"
                },
                "messages": [],
                "meta": {
                    "label": {
                        "id": 1040001,
                        "text": "Sign up",
                        "type": "info"
                    }
                }
            }
        ]
    },
    "organization_id": null,
    "state": "choose_method"
}
```

Response Header:

```
set-cookie: csrf_token_6c0cb2d8e17be2aeca168648961b34f465bed2ffa37dd7fa2ad8b4867a3b1f15
```

2. Submit the registration flow with both information you obtained from the step #1 and user input

   Request

   - Header `Cookie`: obtain from the `set-cookie` header in the response header in step #1
   - Query `flow`: obtain from the `id` attribute in the response body in step #1
   - Body `traits.email`: the account email
   - Body `password`: the account password
   - Body `csrf_token`: the CSRF token that is obtained from the `ui.nodes[0].attributes.value` in the response body in step #1

```
curl --location 'https://kratos.example.com/self-service/registration?flow=eb2bfb29-7fe
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Accept: application/json' \
--header 'Cookie: csrf_token_6c0cb2d8e17be2aeca168648961b34f465bed2ffa37dd7fa2ad8b4867a
--data-urlencode 'traits.email=gyh55799@nezid.com' \
--data-urlencode 'password=1password' \
--data-urlencode 'method=password' \
--data-urlencode 'csrf_token=tmqMMK+4Ii9vIBzA9Xr5DzajAIaIXoyGKecGRbVQDde3cWugSjH+m2bXEw
```

   Response Body


```
{
    "session": {
        "id": "9db7bc79-66ab-4bfe-98e0-9f55cf761318",
        "active": true,
        "expires_at": "2023-12-21T09:42:28.836782153Z",
        "authenticated_at": "2023-12-20T09:42:28.836788985Z",
        "authenticator_assurance_level": "aal1",
        "authentication_methods": [
            {
                "method": "password",
                "aal": "aal1",
                "completed_at": "2023-12-20T09:42:28.83678172Z"
            }
        ],
        "issued_at": "2023-12-20T09:42:28.836782153Z",
        "identity": {
            "id": "d95ad291-d27b-4eaa-89bb-4c151c757ae6",
            "schema_id": "default",
            "schema_url": "https://kratos.example.com/schemas/ZGVmYXVsdA",
            "state": "active",
```

```
            "state_changed_at": "2023-12-20T09:42:28.832395601Z",
            "traits": {
                "email": "gyh55799@nezid.com"
            },
            "verifiable_addresses": [
                {
                    "id": "d25a17e9-c624-46be-a0c1-5002b3495c8a",
                    "value": "gyh55799@nezid.com",
                    "verified": false,
                    "via": "email",
                    "status": "sent",
                    "created_at": "2023-12-20T09:42:28.834034Z",
                    "updated_at": "2023-12-20T09:42:28.834034Z"
                }
            ],
            "recovery_addresses": [
                {
                    "id": "f45a9abe-9ef3-4761-8936-128e7e3616dc",
                    "value": "gyh55799@nezid.com",
                    "via": "email",
                    "created_at": "2023-12-20T09:42:28.834411Z",
                    "updated_at": "2023-12-20T09:42:28.834411Z"
                }
            ],
            "metadata_public": null,
            "created_at": "2023-12-20T09:42:28.833549Z",
            "updated_at": "2023-12-20T09:42:28.833549Z",
            "organization_id": null
        },
        "devices": [
            {
                "id": "0bcffed3-2f65-49d5-87ee-5b401cdfa4a3",
                "ip_address": "10.42.0.1",
                "user_agent": "PostmanRuntime/7.36.0",
                "location": "VN"
            }
        ]
    },
    "identity": {
        "id": "d95ad291-d27b-4eaa-89bb-4c151c757ae6",
        "schema_id": "default",
        "schema_url": "https://kratos.example.com/schemas/ZGVmYXVsdA",
        "state": "active",
        "state_changed_at": "2023-12-20T09:42:28.832395601Z",
        "traits": {
            "email": "gyh55799@nezid.com"
```

```
            },
            "verifiable_addresses": [
                {
                    "id": "d25a17e9-c624-46be-a0c1-5002b3495c8a",
                    "value": "gyh55799@nezid.com",
                    "verified": false,
                    "via": "email",
                    "status": "sent",
                    "created_at": "2023-12-20T09:42:28.834034Z",
                    "updated_at": "2023-12-20T09:42:28.834034Z"
                }
            ],
            "recovery_addresses": [
                {
                    "id": "f45a9abe-9ef3-4761-8936-128e7e3616dc",
                    "value": "gyh55799@nezid.com",
                    "via": "email",
                    "created_at": "2023-12-20T09:42:28.834411Z",
                    "updated_at": "2023-12-20T09:42:28.834411Z"
                }
            ],
            "metadata_public": null,
            "created_at": "2023-12-20T09:42:28.833549Z",
            "updated_at": "2023-12-20T09:42:28.833549Z",
            "organization_id": null
        },
        "continue_with": [
            {
                "action": "show_verification_ui",
                "flow": {
                    "id": "f1b62e42-e311-4222-95cf-bd1f63580cfd",
                    "verifiable_address": "gyh55799@nezid.com",
                    "url": "https://auth.example.com/verification?flow=f1b62e42-e311-4222-9
                }
            }
        ]
}
```


Response Header

```
set-cookie: csrf_token_6c0cb2d8e17be2aeca168648961b34f465bed2ffa37dd7fa2ad8b4867a3b1f15
set-cookie: ory_kratos_session=MTcwMzA2NTM0OHxGSXMzTTBxUzFiNkVHbkU5M3VZY2RvQVhWN0tQZnVh
```

In the response body we got the `continue_with` attribute that indicates the registration flow need performing verification step with new flow id

3. Get the verification flow

An email will be sent to the registration account email, user need to check their inbox and obtain the verification code and back to our UI to complete the verification process. So you need to retrieve the verification flow with the `continue_with[0].flow.id` attribute you received at step #2 to render verification components

Request

- Query `id`: obtain from `continue_with[0].flow.id` attribute in response body at step #2

```
curl --location 'https://kratos.example.com/self-service/verification/flows?id=f1b62e42
--header 'Accept: application/json' \
--header 'Cookie: csrf_token_6c0cb2d8e17be2aeca168648961b34f465bed2ffa37dd7fa2ad8b4867a
```

Response Body


```
{
    "id": "f1b62e42-e311-4222-95cf-bd1f63580cfd",
    "type": "browser",
    "expires_at": "2023-12-20T10:42:28.839397Z",
    "issued_at": "2023-12-20T09:42:28.839397Z",
    "request_url": "https://kratos.example.com/self-service/registration/browser",
    "active": "code",
    "ui": {
        "action": "https://kratos.example.com/self-service/verification?flow=f1b62e42-e
        "method": "POST",
        "nodes": [
            {
                "type": "input",
                "group": "code",
                "attributes": {
                    "name": "method",
                    "type": "hidden",
                    "value": "code",
                    "disabled": false,
                    "node_type": "input"
                },
                "messages": [],
                "meta": {}
            },
            {
                "type": "input",
                "group": "code",
                "attributes": {
                    "name": "code",
                    "type": "text",
                    "required": true,
```

```
                "disabled": false,
                "node_type": "input"
            },
            "messages": [],
            "meta": {
                "label": {
                    "id": 1070011,
                    "text": "Verification code",
                    "type": "info"
                }
            }
        },
        {
            "type": "input",
            "group": "code",
            "attributes": {
                "name": "method",
                "type": "submit",
                "value": "code",
                "disabled": false,
                "node_type": "input"
            },
            "messages": [],
            "meta": {
                "label": {
                    "id": 1070005,
                    "text": "Submit",
                    "type": "info"
                }
            }
        },
        {
            "type": "input",
            "group": "default",
            "attributes": {
                "name": "csrf_token",
                "type": "hidden",
                "value": "vbeh+FEMpPO5pIicduwM9ZY77Dvmt/0CodHzW+tCd+tnAfpw6e2RMm2jX
                "required": true,
                "disabled": false,
                "node_type": "input"
            },
            "messages": [],
            "meta": {}
        }
    ],
```

```
        "messages": [
            {
                "id": 1080003,
                "text": "An email containing a verification code has been sent to the e
                "type": "info"
            }
        ]
    },
    "state": "sent_email"
}
```

4. Verify the registration account email

   In this example, we are going to use the code `489496` to demonstrate the flow

   Request

   - Query `flow`: obtain from the `continue_with[0].flow.id` attribute in the response body in step #2 (value: `44db86dc-131f-47c6-ae2b-38981a686d45`)
   - Body `code`: User input

```
curl --location 'https://kratos.example.com/self-service/verification?flow=f1b62e42-e31
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Accept: application/json' \
--header 'Cookie: csrf_token_6c0cb2d8e17be2aeca168648961b34f465bed2ffa37dd7fa2ad8b4867a
--data-urlencode 'code=489496' \
--data-urlencode 'csrf_token=vbeh+FEMpPO5pIicduwM9ZY77Dvmt/0CodHzW+tCd+tnAfpw6e2RMm2jXB
```

   Response Body

```
{
    "id": "f1b62e42-e311-4222-95cf-bd1f63580cfd",
    "type": "browser",
    "expires_at": "2023-12-20T10:42:28.839397Z",
    "issued_at": "2023-12-20T09:42:28.839397Z",
    "request_url": "https://kratos.example.com/self-service/registration/browser",
    "active": "code",
    "ui": {
        "action": "https://auth.example.com/",
        "method": "GET",
        "nodes": [
            {
                "type": "input",
                "group": "default",
                "attributes": {
                    "name": "csrf_token",
                    "type": "hidden",
                    "value": "XbFBP/PHARIDAaxTKsfj/bWUOO/OHRTVSPhZOH99fNKHBxq3SyY009cGe
```

```
                        "required": true,
                        "disabled": false,
                        "node_type": "input"
                    },
                    "messages": [],
                    "meta": {}
                },
                {
                    "type": "a",
                    "group": "code",
                    "attributes": {
                        "href": "https://auth.example.com/",
                        "title": {
                            "id": 1070009,
                            "text": "Continue",
                            "type": "info"
                        },
                        "id": "continue",
                        "node_type": "a"
                    },
                    "messages": [],
                    "meta": {
                        "label": {
                            "id": 1070009,
                            "text": "Continue",
                            "type": "info"
                        }
                    }
                }
            ],
            "messages": [
                {
                    "id": 1080002,
                    "text": "You successfully verified your email address.",
                    "type": "success"
                }
            ]
        },
        "state": "passed_challenge"
}
```


Login

1. Init the login flow

   Request

```
curl --location 'https://kratos.example.com/self-service/login/browser?refresh=true' \
--header 'Accept: application/json' \
```

Response Body


```
{
    "id": "04d459df-5c63-44eb-830a-665eeeaed215",
    "organization_id": null,
    "type": "browser",
    "expires_at": "2023-12-20T13:21:15.823403688Z",
    "issued_at": "2023-12-20T13:11:15.823403688Z",
    "request_url": "https://kratos.example.com/self-service/login/browser?refresh=true"
    "ui": {
        "action": "https://kratos.example.com/self-service/login?flow=04d459df-5c63-44e
        "method": "POST",
        "nodes": [
            {
                "type": "input",
                "group": "default",
                "attributes": {
                    "name": "csrf_token",
                    "type": "hidden",
                    "value": "I2PGDhh/vNT/9W0wunEMViu8dYDvgSJvlCqWvQHxAafRwkxG9G0rVCVyC
                    "required": true,
                    "disabled": false,
                    "node_type": "input"
                },
                "messages": [],
                "meta": {}
            },
            {
                "type": "input",
                "group": "default",
                "attributes": {
                    "name": "identifier",
                    "type": "text",
                    "value": "",
                    "required": true,
                    "disabled": false,
                    "node_type": "input"
                },
                "messages": [],
                "meta": {
                    "label": {
                        "id": 1070002,
                        "text": "E-Mail",
                        "type": "info",
```

```
                "context": {
                    "title": "E-Mail"
                }
            }
        }
    },
    {
        "type": "input",
        "group": "password",
        "attributes": {
            "name": "password",
            "type": "password",
            "required": true,
            "autocomplete": "current-password",
            "disabled": false,
            "node_type": "input"
        },
        "messages": [],
        "meta": {
            "label": {
                "id": 1070001,
                "text": "Password",
                "type": "info"
            }
        }
    },
    {
        "type": "input",
        "group": "password",
        "attributes": {
            "name": "method",
            "type": "submit",
            "value": "password",
            "disabled": false,
            "node_type": "input"
        },
        "messages": [],
        "meta": {
            "label": {
                "id": 1010001,
                "text": "Sign in",
                "type": "info"
            }
        }
    }
]
```

```
    },
    "created_at": "2023-12-20T13:11:15.825857Z",
    "updated_at": "2023-12-20T13:11:15.825857Z",
    "refresh": false,
    "requested_aal": "aal1",
    "state": "choose_method"
}
```

2. Submit the login flow with both information you obtained from the step #1 and user credentials

   Request

   - Query `flow`: obtain from the `id` attribute in the response body in step #1
   - Body `identifier`: the account email
   - Body `password`: the account password
   - Body `csrf_token`: the CSRF token that is obtained from the `ui.nodes[0].attributes.value` in the response body in step #1

```
curl --location 'https://kratos.example.com/self-service/login?flow=04d459df-5c63-44eb-
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Accept: application/json' \
--header 'Cookie: csrf_token_6c0cb2d8e17be2aeca168648961b34f465bed2ffa37dd7fa2ad8b4867a
--data-urlencode 'method=password' \
--data-urlencode 'password=1password' \
--data-urlencode 'identifier=gyh55799@nezid.com' \
--data-urlencode 'csrf_token=I2PGDhh/vNT/9W0wunEMViu8dYDvgSJvlCqWvQHxAafRwkxG9G0rVCVyCs
```

   Response Body

```
{
    "session": {
        "id": "468ca40c-ae5b-4037-8914-f65597626c3d",
        "active": true,
        "expires_at": "2023-12-21T13:13:35.413927139Z",
        "authenticated_at": "2023-12-20T13:13:35.413927139Z",
        "authenticator_assurance_level": "aal1",
        "authentication_methods": [
            {
                "method": "password",
                "aal": "aal1",
                "completed_at": "2023-12-20T13:13:35.413921865Z"
            }
        ],
        "issued_at": "2023-12-20T13:13:35.413927139Z",
        "identity": {
            "id": "d95ad291-d27b-4eaa-89bb-4c151c757ae6",
            "schema_id": "default",
```

```
            "schema_url": "https://kratos.example.com/schemas/ZGVmYXVsdA",
            "state": "active",
            "state_changed_at": "2023-12-20T09:42:28.832395Z",
            "traits": {
                "email": "gyh55799@nezid.com"
            },
            "verifiable_addresses": [
                {
                    "id": "d25a17e9-c624-46be-a0c1-5002b3495c8a",
                    "value": "gyh55799@nezid.com",
                    "verified": true,
                    "via": "email",
                    "status": "completed",
                    "verified_at": "2023-12-20T09:46:06.984885Z",
                    "created_at": "2023-12-20T09:42:28.834034Z",
                    "updated_at": "2023-12-20T09:42:28.834034Z"
                }
            ],
            "recovery_addresses": [
                {
                    "id": "f45a9abe-9ef3-4761-8936-128e7e3616dc",
                    "value": "gyh55799@nezid.com",
                    "via": "email",
                    "created_at": "2023-12-20T09:42:28.834411Z",
                    "updated_at": "2023-12-20T09:42:28.834411Z"
                }
            ],
            "metadata_public": null,
            "created_at": "2023-12-20T09:42:28.833549Z",
            "updated_at": "2023-12-20T09:42:28.833549Z",
            "organization_id": null
        },
        "devices": [
            {
                "id": "0ea9cb85-b617-4dc3-b244-920e3df23522",
                "ip_address": "10.42.0.1",
                "user_agent": "PostmanRuntime/7.36.0",
                "location": "VN"
            }
        ]
    }
}
```


Response Header


```
set-cookie: csrf_token_6c0cb2d8e17be2aeca168648961b34f465bed2ffa37dd7fa2ad8b4867a3b1f15
set-cookie: ory_kratos_session=MTcwMzA3ODAxNXx2QU1sR1ZzNWJhX1gtXzJZa1Vtck9NbVVxWklVOWJZ
```

3. In further requests, you need to sent a request the header `Cookie` with a value from `set-cookie` that has value `ory_kratos_session`

Logout An example of how to use `Cookie` header is logout the session. Lets do it

1. Init the logout flow

   In this API request, I am going to attach the header `Cookie` with the header value of `set-cookie` we got in Login step #2

   Request:

   ```
   curl --location 'https://kratos.example.com/self-service/logout/browser' \
   --header 'Cookie:  ory_kratos_session=MTcwMzA3ODAxNXx2QU1sR1ZzNWJhX1gtXzJZa1Vtck9NbVVxW
   --header 'Accept: application/json'
   ```

   Respond body

   ```
   {
       "logout_url": "https://kratos.example.com/self-service/logout?token=ory_lo_1vwkCcHG
       "logout_token": "ory_lo_1vwkCcHGln6jTQ6aX4AbU8ueCaU7UkyC"
   }
   ```

2. Redirect user to the url of attribute `logout_url` in the response body at step #1

Whoami Whoami is an API that return the information of current session of user. The returning response may contains session, authenitcation, indentity and device information.

Request

```
curl --location 'https://kratos.example.com/sessions/whoami' \
--header 'Accept: application/json' \
--header 'Cookie: csrf_token_6c0cb2d8e17be2aeca168648961b34f465bed2ffa37dd7fa2ad8b4867a3b1f1
```

Response Body

```
{
    "id": "468ca40c-ae5b-4037-8914-f65597626c3d",
    "active": true,
    "expires_at": "2023-12-21T13:13:35.413927Z",
    "authenticated_at": "2023-12-20T13:13:35.413927Z",
    "authenticator_assurance_level": "aal1",
    "authentication_methods": [
        {
            "method": "password",
            "aal": "aal1",
            "completed_at": "2023-12-20T13:13:35.413921865Z"
        }
```

```
    ],
    "issued_at": "2023-12-20T13:13:35.413927Z",
    "identity": {
        "id": "d95ad291-d27b-4eaa-89bb-4c151c757ae6",
        "schema_id": "default",
        "schema_url": "https://kratos.example.com/schemas/ZGVmYXVsdA",
        "state": "active",
        "state_changed_at": "2023-12-20T09:42:28.832395Z",
        "traits": {
            "email": "gyh55799@nezid.com"
        },
        "verifiable_addresses": [
            {
                "id": "d25a17e9-c624-46be-a0c1-5002b3495c8a",
                "value": "gyh55799@nezid.com",
                "verified": true,
                "via": "email",
                "status": "completed",
                "verified_at": "2023-12-20T09:46:06.984885Z",
                "created_at": "2023-12-20T09:42:28.834034Z",
                "updated_at": "2023-12-20T09:42:28.834034Z"
            }
        ],
        "recovery_addresses": [
            {
                "id": "f45a9abe-9ef3-4761-8936-128e7e3616dc",
                "value": "gyh55799@nezid.com",
                "via": "email",
                "created_at": "2023-12-20T09:42:28.834411Z",
                "updated_at": "2023-12-20T09:42:28.834411Z"
            }
        ],
        "metadata_public": null,
        "created_at": "2023-12-20T09:42:28.833549Z",
        "updated_at": "2023-12-20T09:42:28.833549Z",
        "organization_id": null
    },
    "devices": [
        {
            "id": "0ea9cb85-b617-4dc3-b244-920e3df23522",
            "ip_address": "10.42.0.1",
            "user_agent": "PostmanRuntime/7.36.0",
            "location": "VN"
        }
    ]
}
```

Response Header

```
x-kratos-authenticated-identity-id: d95ad291-d27b-4eaa-89bb-4c151c757ae6
```