Cheat sheet

Red Hat OpenShift Application Services

The Red Hat OpenShift Application Services command-line interface (CLI) `rhoas` lets you manage your application services from a terminal. To install it, review the Installing and configuring the rhoas CLI page. For more details and the latest information, check the GitHub repository.

Login commands

```
rhoas login
```

Log in to `rhoas`.

```
rhoas login
```

Log in to `rhoas`.

Apache Kafka management

Manage and interact with Kafka instances, Kafka topics, Consumer Groups, and Access Control Lists (ACL). Each Kafka service includes a full Apache Kafka cluster, bootstrap servers, and the configurations needed to connect to producer and consumer services.

```
rhoas kafka create --name my-kafka-instance
```

Create a Kafka instance.

```
rhoas kafka describe --name my-kafka-instance
```

View configuration details of a Kafka instance.

```
rhoas kafka list
```

List all Kafka instances.

```
rhoas kafka update --name my-kafka-instance [flags]
```

Update configuration details for a Kafka instance.

```
rhoas kafka use --name my-kafka-instance
```

Set the current Kafka instance.

```
rhoas kafka delete --name my-kafka-instance
```

Delete a Kafka instance.

Apache Kafka topic management

This section covers commands for managing Kafka topics in the current Kafka instance.

```
rhoas kafka topic list
```

List all topics.

```
rhoas kafka topic create --name my-topic
```

Create a Kafka topic.

```
rhoas kafka topic describe --name my-topic
```

Describe a Kafka topic.

```
rhoas kafka topic update --name my-topic --retention-ms -1
```

Update configuration details for a Kafka topic.

```
rhoas kafka topic consume --name my-topic --partition 0 --wait
```

Consume messages from a Kafka topic.

```
rhoas kafka topic produce --name my-topic --file="message.json"
```

Produce a new message to a Kafka topic.

```
rhoas kafka topic delete --name my-topic
```

Delete a Kafka topic.

Apache Kafka Consumer Group Management

This section covers commands for managing Kafka Consumer Groups in the current Kafka instance.

```
rhoas kafka topic list
```

List all consumer groups.

```
rhoas kafka consumer-group describe --id consumer-group-01
```

View detailed information for a consumer group and its members.

```
rhoas kafka consumer-group reset-offset --id consumer-group-01 --topic my-
topic --offset [earliest|latest|absolute|timestamp]
```

Reset partition offsets for a consumer group in a particular Kafka topic.

```
rhoas kafka consumer-group delete --id consumer-group-01
```

Delete a consumer group.

Apache Kafka Access Control List management

This section covers commands for managing Kafka Access Control Lists (ACLs) for users and service accounts in the current Kafka instance.

```
rhoas kafka acl create --operation all --permission allow --topic my-topic --
user dev-user
```

Create a Kafka ACL rule.

```
rhoas kafka acl list
```

List all Kafka ACL rules.

```
rhoas kafka acl grant-access --producer --consumer --service-account ID --
topic my-topic --group all
```

Add ACL rules to grant users access to produce and consume from topics.

```
rhoas kafka acl grant-admin --service-account ID
```

Grant an account permissions to create and delete ACLs in the Kafka instance.

```
rhoas kafka acl delete --operation all --permission allow --topic my-topic --
service-account ID
```

Delete Kafka ACLs matching the provided filters.

Common flags available:

- `--operation string`: Set the ACL operation. Choose from the following: `all`, `alter`, `alter-configs`, `create`, `delete`, `describe`, `describe-configs`, `read`, `write`
- `-permission string`: Set the ACL permission. Choose from the following: `allow`, `deny`

Service Registry management

Manage and interact with your Service Registry instances directly from the command line.

```
rhoas service-registry create --name my-registry
```

Create a Service Registry instance.

```
rhoas service-registry list
```

List Service Registry instances.

```
rhoas service-registry describe --name my-registry
```

Describe a Service Registry instance.

```
rhoas service-registry use --name my-registry
```

Use a Service Registry instance.

```
rhoas service-registry delete --name my-registry
```

Delete a Service Registry instance.

Service Registry artifact management

This section covers commands for managing Service Registry schema and API artifacts in the current selected Service Registry instance.

```
rhoas service-registry artifact create --artifact-id my-artifact --file my-
artifact.asvc --type AVRO
```

Create new artifact from file or standard input.

```
rhoas service-registry artifact list
```

List artifacts.

```
rhoas service-registry artifact metadata-get --artifact-id my-artifact
```

Get artifact metadata.

```
rhoas service-registry artifact metadata-set --artifact-id my-artifact --
description my-artifact
```

Update artifact metadata.

```
rhoas service-registry artifact download --global-id GLOBAL_ID
```

Download artifacts from Service Registry using global identifiers.

```
rhoas service-registry artifact get --artifact-id my-artifact
```

Get artifact by ID, group, and version.

```
rhoas service-registry artifact update --artifact-id my-artifact --file my-
artifact.asvc
```

Update artifact.

```
rhoas service-registry artifact state-set --artifact-id my-artifact --state
[DISABLED|ENABLED|DEPRECATED]
```

Set artifact state.

```
rhoas service-registry artifact versions --artifact-id my-artifact
```

Get latest artifact versions by artifact-id and group.

```
rhoas service-registry artifact export --output-file=export.zip
```

Export all artifacts and metadata from Service Registry instance.

```
rhoas service-registry artifact import --file=export.zip
```

Import data into a Service Registry instance.

```
rhoas service-registry artifact delete --artifact-id ID
```

Deletes an artifact or all artifacts in a given group.

Service Registry role management

Manage Service Registry roles using a set of commands that give users one of following permissions:

- `viewer` : provides read access
- `manager` : provides read and write access
- `admin` : provides admin access as well as read and write access

```
rhoas service-registry role add --role manager --service-account ID
```

Add or update principal role.

```
rhoas service-registry role list
```

List roles.

```
rhoas service-registry role revoke --service-account ID
```

Revoke role for principal.

Service Registry rule management

Configure the validity and compatibility rules that govern artifact content.

```
rhoas service-registry rule list
```

List the validity and compatibility rules.

```
rhoas service-registry rule enable --rule-type=validity --config=syntax-only
```

Enable validity and compatibility rules.

```
rhoas service-registry rule describe --rule-type=validity
```

Display the configuration details of a rule.

```
rhoas service-registry rule update --rule-type=validity --config=syntax-only
```

Update the configuration of rules.

```
rhoas service-registry rule disable --rule-type=validity
```

Disable validity and compatibility rules.

For more information, see Supported Service Registry content and rules.

Service Registry setting management

Commands for manage settings for a Service Registry instance. The available settings include the following options:

- `registry.auth.authenticated-read-access.enabled`: Specifies whether Service Registry grants at least read-only access to requests from any authenticated user in the same organization, regardless of their user role.
- `registry.auth.basic-auth-client-credentials.enabled`: Specifies whether Service Registry users can authenticate using HTTP basic authentication, in addition to OAuth.
- `registry.auth.owner-only-authorization`: Specifies whether only the user who creates an artifact can modify that artifact.
- `registry.ccompat.legacy-id-mode.enabled`: Specifies whether the Confluent Schema Registry compatibility API uses `globalId` instead of `contentId` as an artifact identifier.

```
rhoas service-registry setting list
```

List all the settings for a Service Registry.

```
rhoas service-registry setting get --name SETTING-NAME
```

Get value of the setting for a Service Registry instance.

```
rhoas service-registry setting set --name SETTING-NAME --value SETTING-VALUE
```

Set value of the setting for a Service Registry instance.

Service account management

Manage service accounts. Service accounts enable you to connect your applications to a Kafka instance.

The credentials can be exported in the following structures:

- `env` (default): Store credentials in an env file as environment variables.
- `json` : Store credentials in a JSON file.
- `properties` : Store credentials in a properties file, which is typically used in Java-related technologies.
- `secret` : Store credentials in a Kubernetes secret file.

```
rhoas service-account create --short-description my-service-account --file-
format env
```

Create a service account with credentials that are saved to a file.

```
rhoas service-account list
```

List all service accounts.

```
rhoas service-account describe --id ID
```

View configuration details for a service account.

```
rhoas service-account reset-credentials --id ID --file-format env
```

Reset service account credentials.

```
rhoas service-account delete --id ID
```

Delete a service account.

Configuration management

Generate configuration files for the service context to connect with various tools and platforms.

You must specify an output format into which the credentials will be stored:

- `env` (default) : Store credentials in an env file as environment variables.
- `json` : Store credentials in a JSON file.
- `properties` : Store credentials in a properties file, which is typically used in Java-related technologies.
- `configmap` : Store configurations in a Kubernetes ConfigMap file.

```
rhoas generate-config --type json
```

Status management

View the status of application services in a service context.

```
rhoas status
```

View the status of all application services in the current service context.

```
rhoas status kafka
```

View the status of all Kafka instances in a specific service context.

```
rhoas status service-registry
```

View the status of all Service Registry services in a specific service context.

```
rhoas generate-config --type json
```

Context management

Group your service instances into reusable contexts. Context can be used when running other `rhoas` commands or to generate service configuration.

A service context is a group of application service instances and their service identifiers. By using service contexts, you can group together application service instances that you want to use together.

```
rhoas context create --name sandbox-dev
```

Create a service context.

```
rhoas context list
```

List service contexts.

```
rhoas context use --name sandbox-dev
```

Set the current context.

```
rhoas context set-kafka --name=my-kafka-instance
```

Set the current Kafka instance.

```
rhoas context set-service-registry --name=my-registry
```

Use a Service Registry instance.

```
rhoas context status --name sandbox-dev
```

View the status of application services in a service context.

```
rhoas context unset --name sandbox-dev --services service-registry
```

Unset services in context.

```
rhoas context delete --name sandbox-dev
```

Permanently delete a service context.

Miscellaneous

```
rhoas whoami
```

View the username of the current user.

```
rhoas authtoken
```

View the authentication token of the current user.

```
rhoas version
```

Display version information about the rhoas client.

```
rhoas help command
```

Display help about a command.

Enabling command completion

On Bash:

```
rhoas completion bash > rhoas_completions
sudo mv rhoas_completions /etc/bash_completion.d/rhoas
```

On zsh:

```
rhoas completion zsh > "${fpath[1]}/_rhoas"
echo "autoload -U compinit; compinit" >> ~/.zshrc
```

On Fish:

```
rhoas completion fish > ~/.config/fish/completions/rhoas.fish
```

Global flags

Most of the commands includes the following global flags:

- `-o`, `--output string`: Specify the output format: `json`, `yaml`, `yml`
- `-h`, `--help`: Show help for a command.
- `-v`, `--verbose`: Enable verbose mode.

Full command API

The full list of commands, with options, flags, and examples is available from the App Services CLI website.