

---

# Backend.AI Documentation

*Release 24.03.3*

**Lablup Inc.**

**May 13, 2024**



# CONTENTS

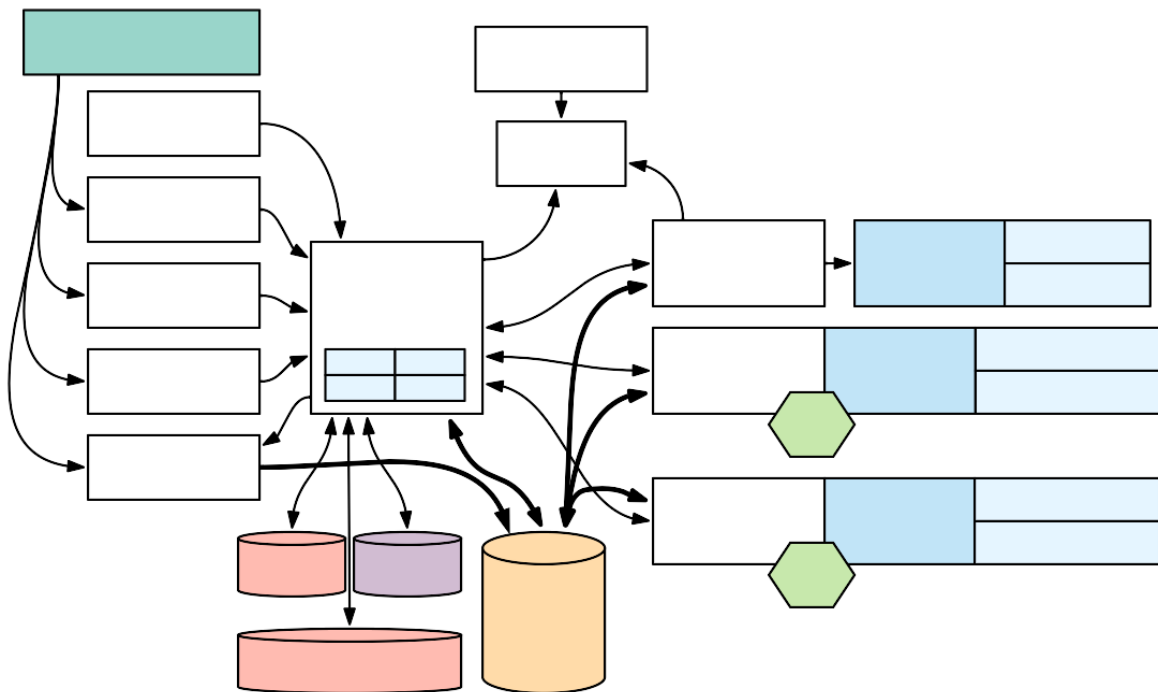


”

,



## BACKEND.AI CONCEPTS



### 1.1 Service Components

#### 1.1.1 Public-facing services

Manager and Webserver

App Proxy

Storage Proxy

,

FastTrack (Enterprise only)

## 1.1.2 Resource Management

Sokovan Orchestrator

Agent

“”

## 1.1.3 Internal services

Event bus



Control Panel (Enterprise only)

Forklift (Enterprise only)

Reservoir (Enterprise only)

Container Registry

,

## 1.2 Computing

### 1.2.1 Sessions and kernels

“”.

,

#### Kernel roles in a cluster session

“” “” “”, “”, “”, ..., “” “”

“” “”

### 1.2.2 Session templates

### 1.2.3 Session types

Feature	Compute (Interactive)	Compute (Batch)	Inference	System
Code execution	✓			
Service port	✓	✓	✓	✓
Dependencies		✓		
Session result		✓		
Clustering	✓	✓	✓	✓

Interactive compute session

Batch compute session

“”

Dependencies between compute sessions

Inference session

System session

## 1.2.4 Scheduling

Session selection strategy

Heuristic FIFO

“”

Dominant resource fairness (DRF)

Agent selection strategy

Concentrated

Dispersed

Custom

## 1.3 Resource Management

### 1.3.1 Resource slots

“”.

Resource slot name	Device name	Slot name

Slot type	Meaning	Examples
	“”	

### 1.3.2 Compute plugins

### 1.3.3 Resource groups

## 1.4 User Management

### 1.4.1 Users

,

### 1.4.2 Projects

## 1.5 Cluster Networking

### 1.5.1 Single-node cluster session

### 1.5.2 Multi-node cluster session

### 1.5.3 Detection of clustered setups

,

Environment Variable	Meaning	Examples

## 1.6 Storage Management

### 1.6.1 Virtual folders

“” “”

User-owned vfolders

Project-owned vfolders

---

---

VFolder invitations and permissions

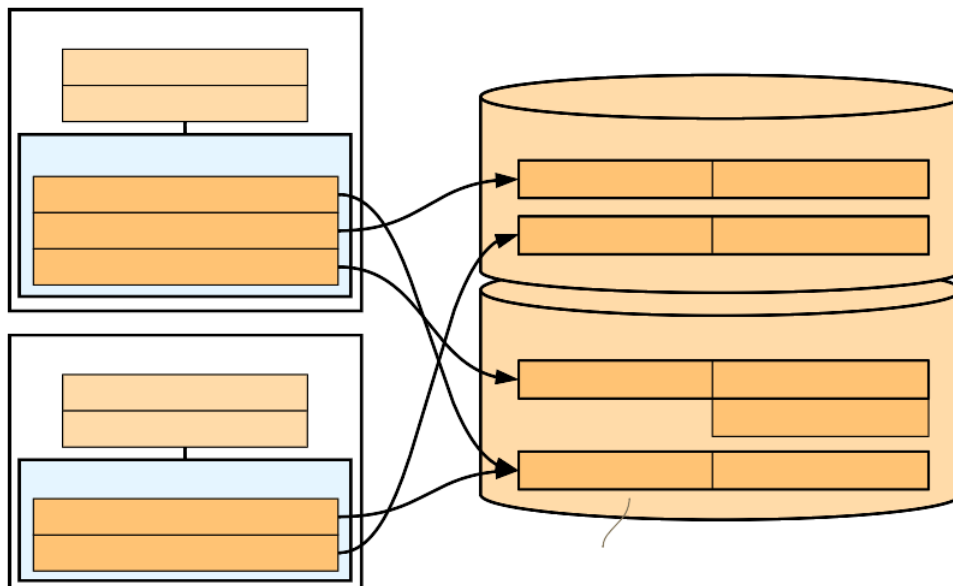
Volume-level permissions

Auto-mount vfolders

,

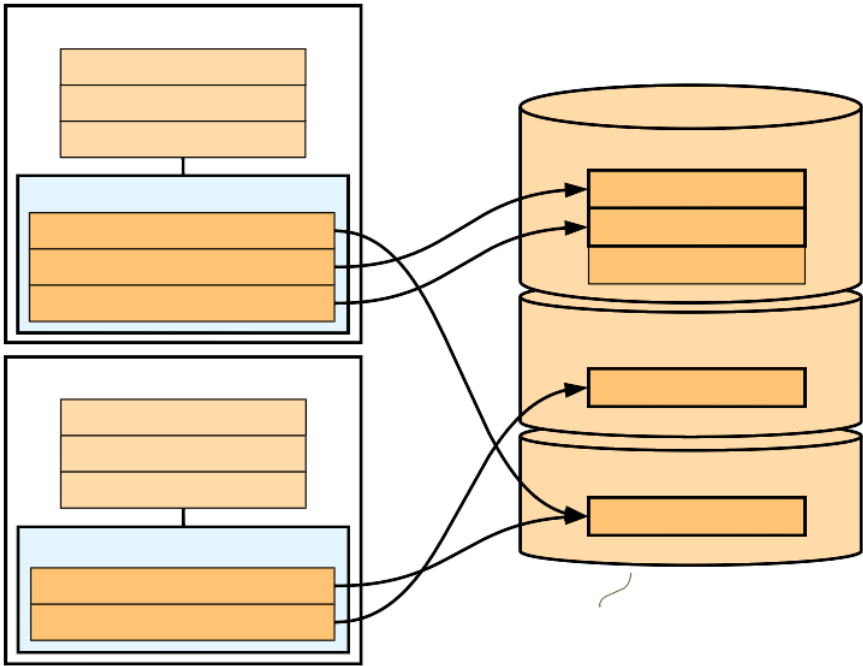
## 1.6.2 Quota scopes

Storage with per-directory quota



,

Storage with per-volume quota



,

,

## 1.7 Configuration

### 1.7.1 Shared config

1.7.2 Local config

1.8 Monitoring

1.8.1 Dashboard (Enterprise only)

1.8.2 Alerts (Enterprise only)

1.9 FAQ

vs. Notebooks

Product	Role	Value

vs. Orchestration Frameworks

Product	Target	Value

vs. Big-data and AI Frameworks

Product	Role	Value





## INSTALLATION GUIDES

### 2.1 Install from Source

---

---

#### 2.1.1 Setting Up Manager and Agent (single node, all-in-one)

#### 2.1.2 Setting Up Additional Agents (multi-node)

Updating manager configuration for multi-nodes

Installing additional agents in different nodes

“”

---

---

### 2.1.3 Setting Up Accelerators

,

### 2.1.4 Setting Up Shared Storage

,

### 2.1.5 Configuring Overlay Networks for Multi-node Training (Optional)

---

---

,

## 2.2 Install from Packages

,

## 2.2.1 Setup OS Environment

---

---

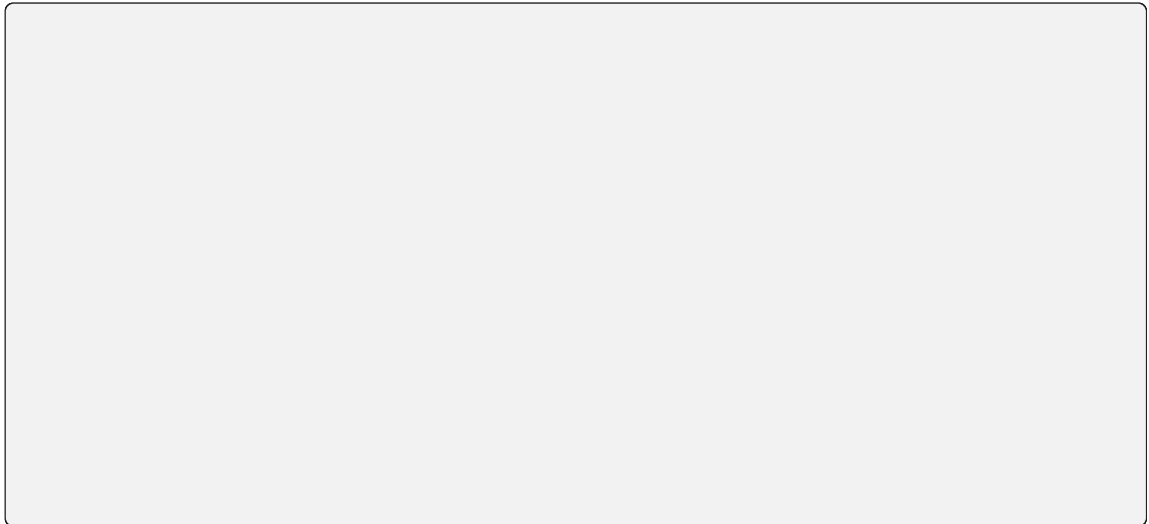
Create a user account for operation

Install Docker engine

Optimize sysctl/ulimit parameters

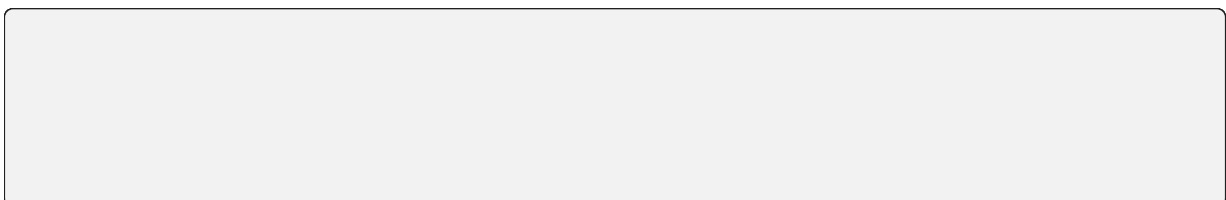
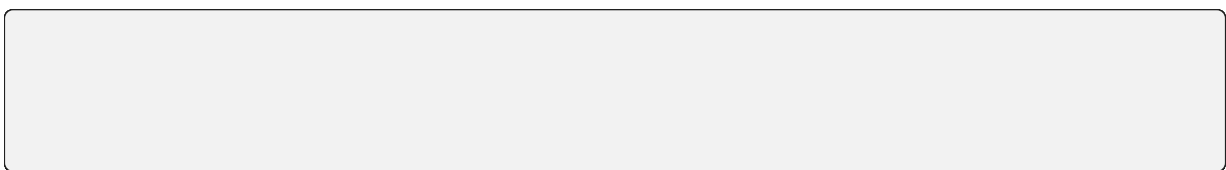
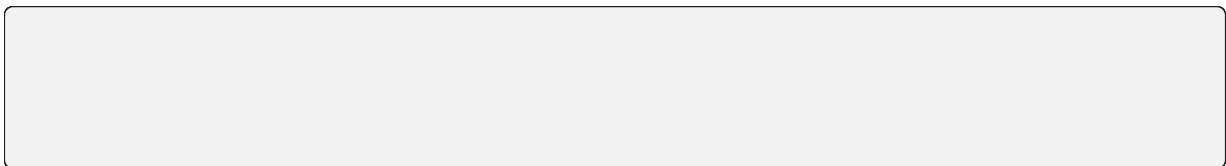
---

---



## Prepare required Python versions and virtual environments

### Use a standalone static built Python (Recommended)



**(Alternative) Use pyenv to manually build and select a specific Python version**

**Configure network aliases**

**Mount a shared storage**

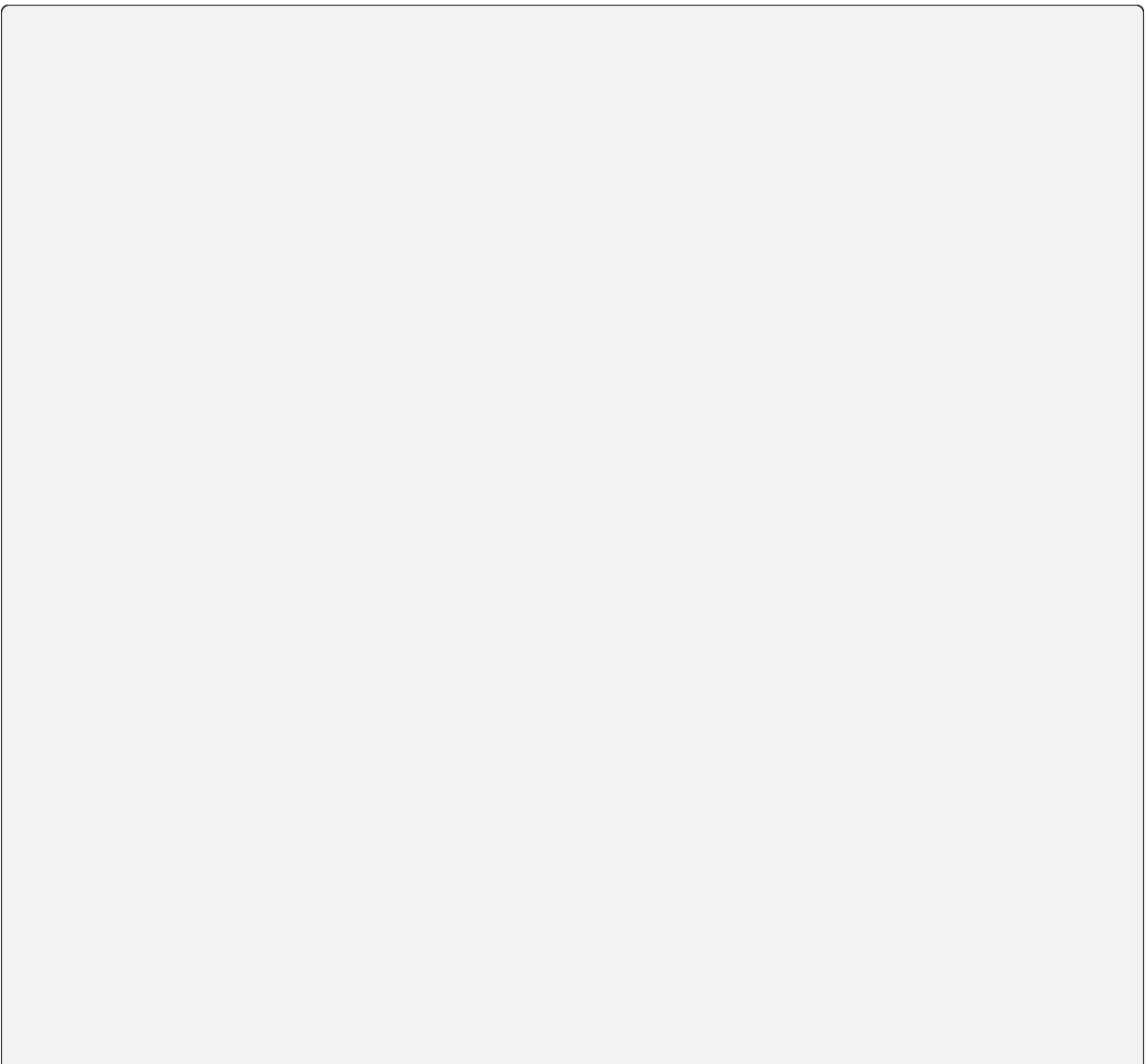
Setup accelerators

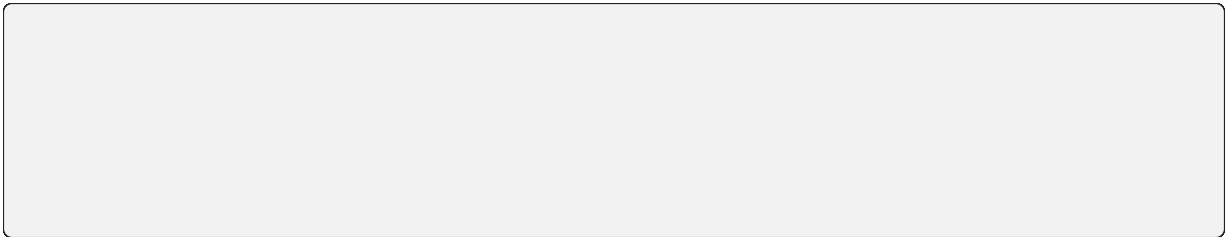
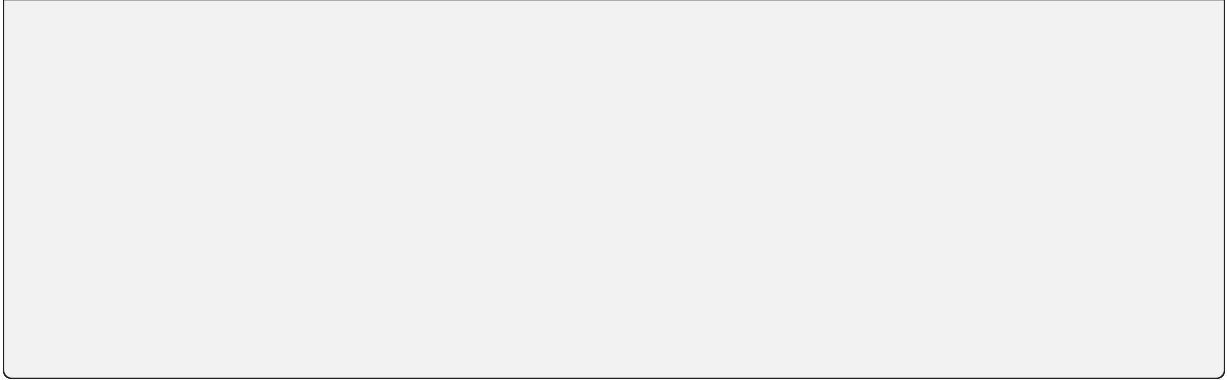
Pull container images



### 2.2.2 Prepare Database

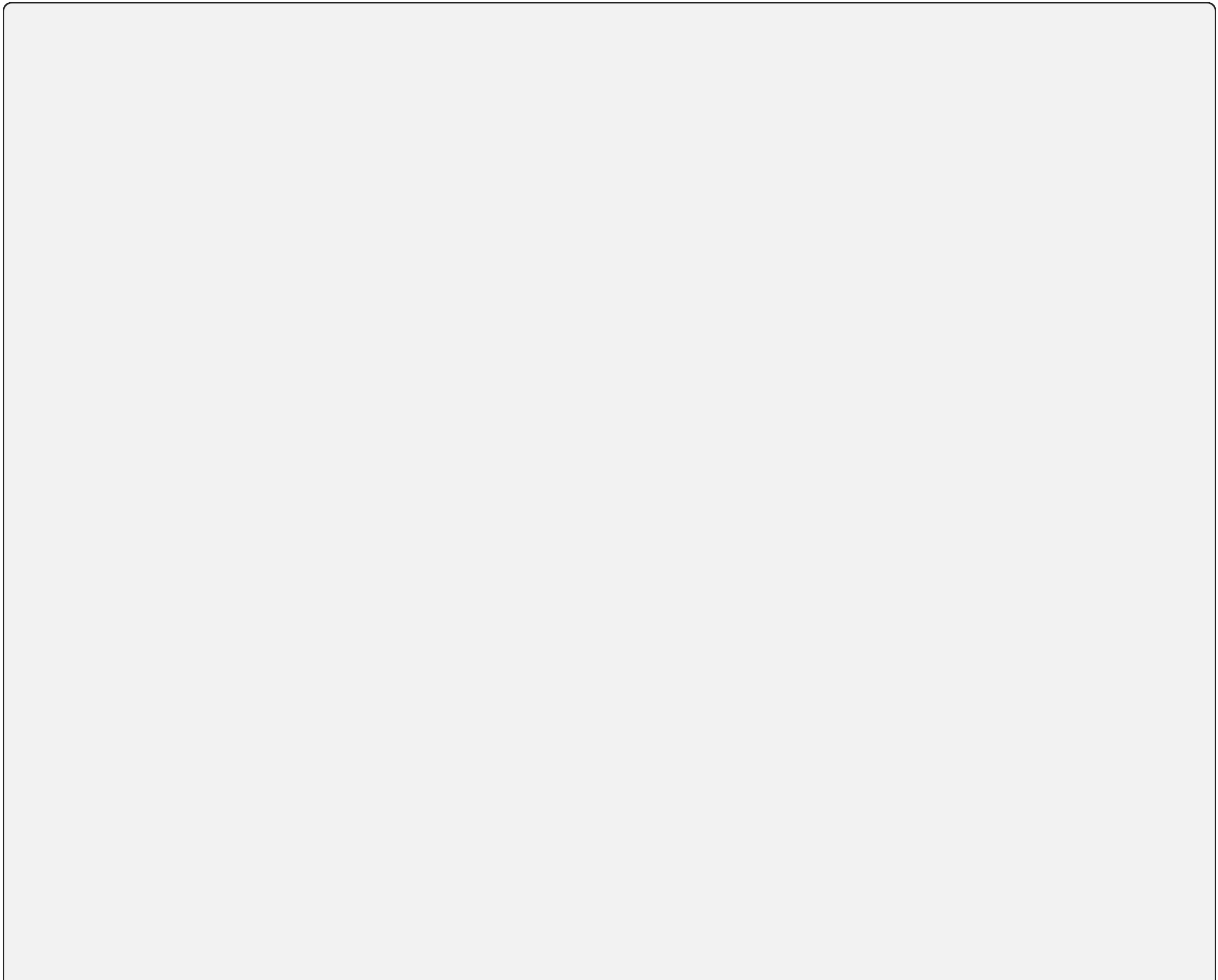
,





### 2.2.3 Prepare Cache Service

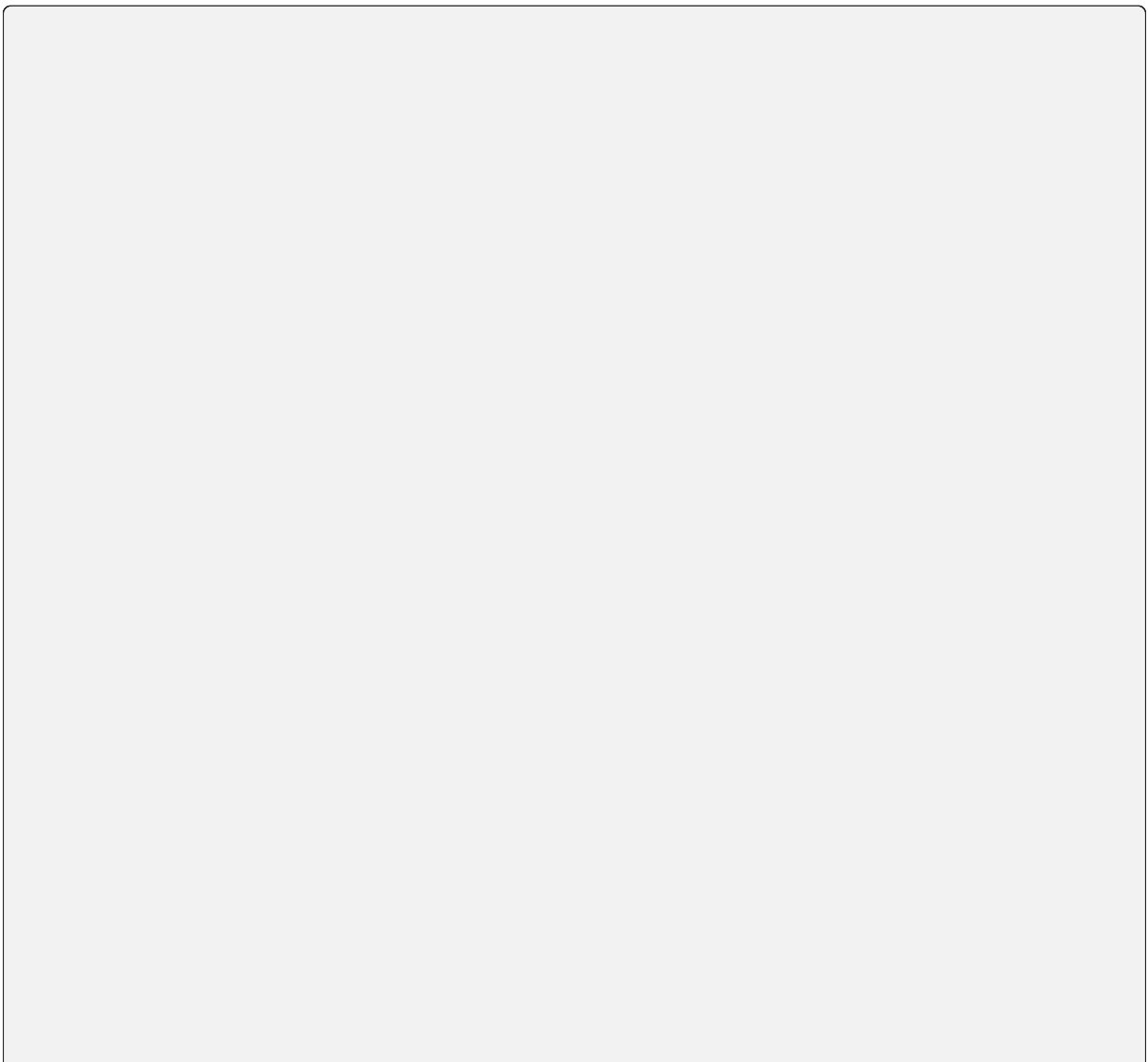
,



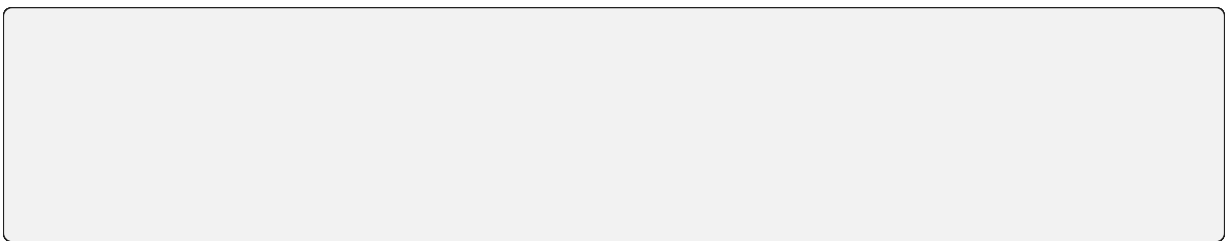
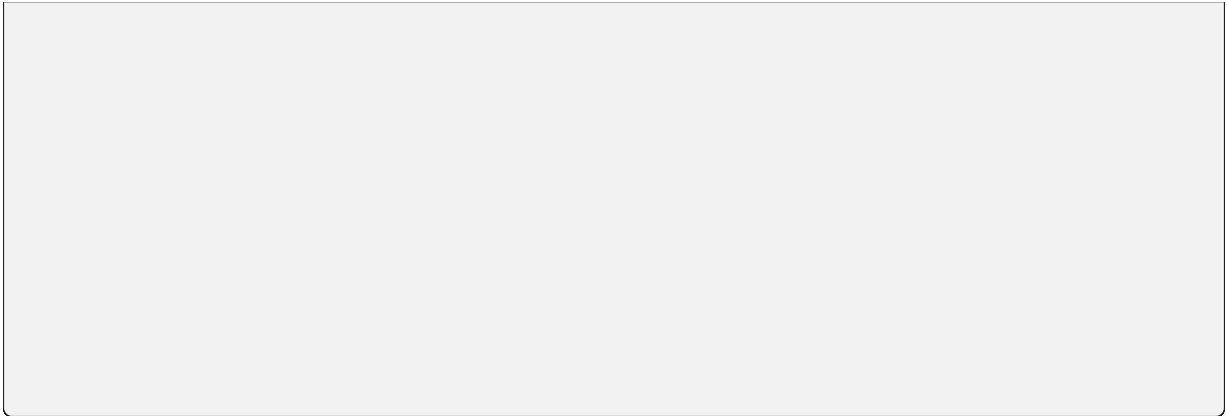


## 2.2.4 Prepare Config Service

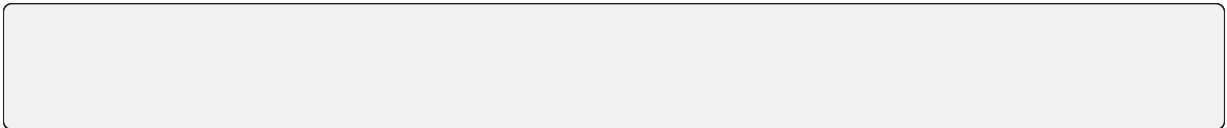
,



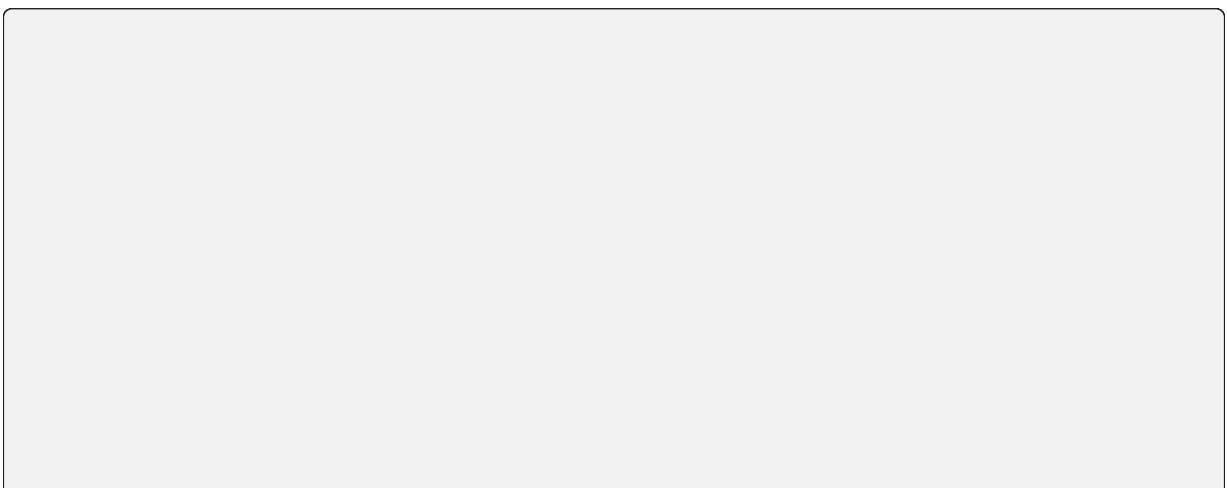


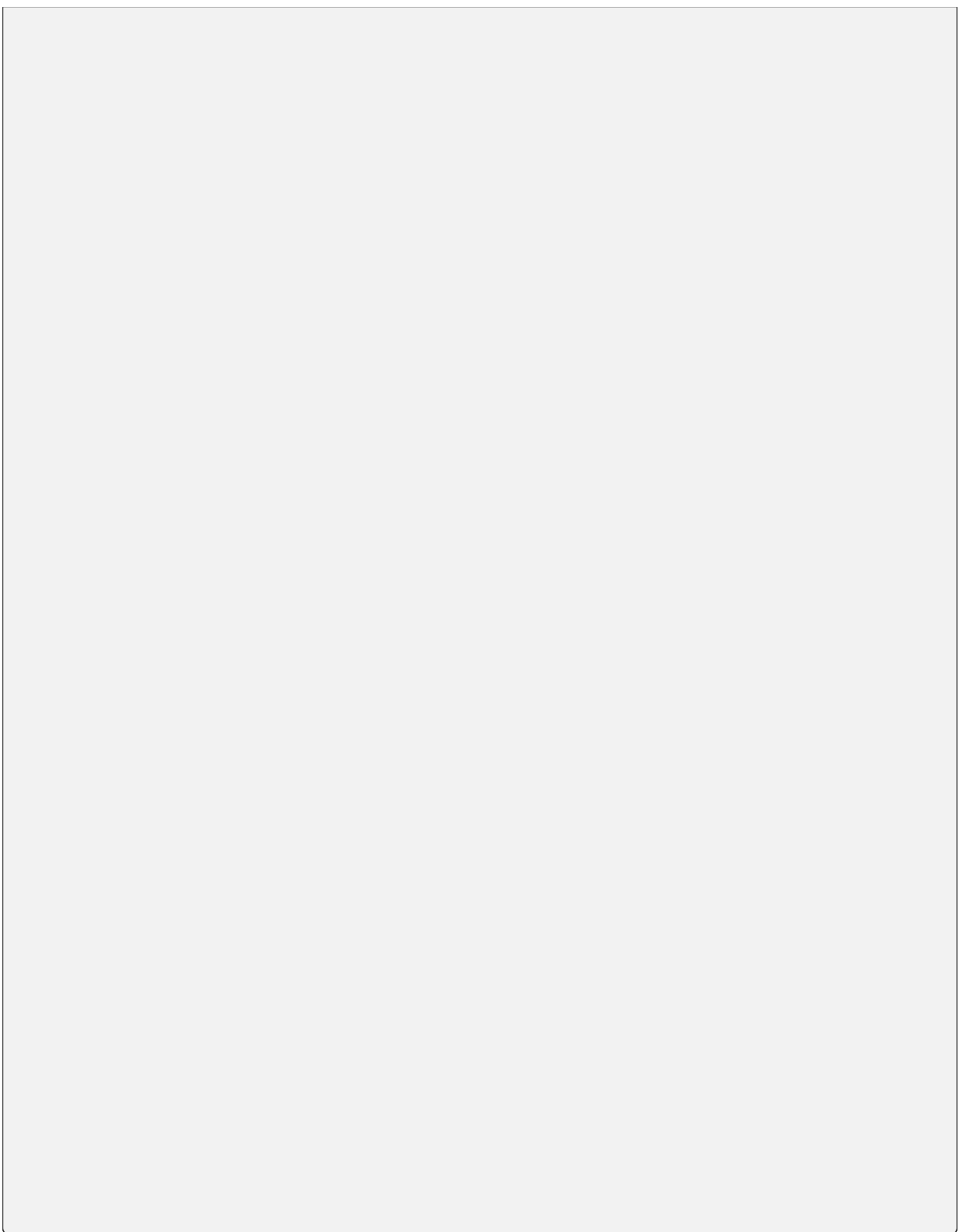


### 2.2.5 Install Backend.AI Manager



#### Local configuration





Global configuration

,

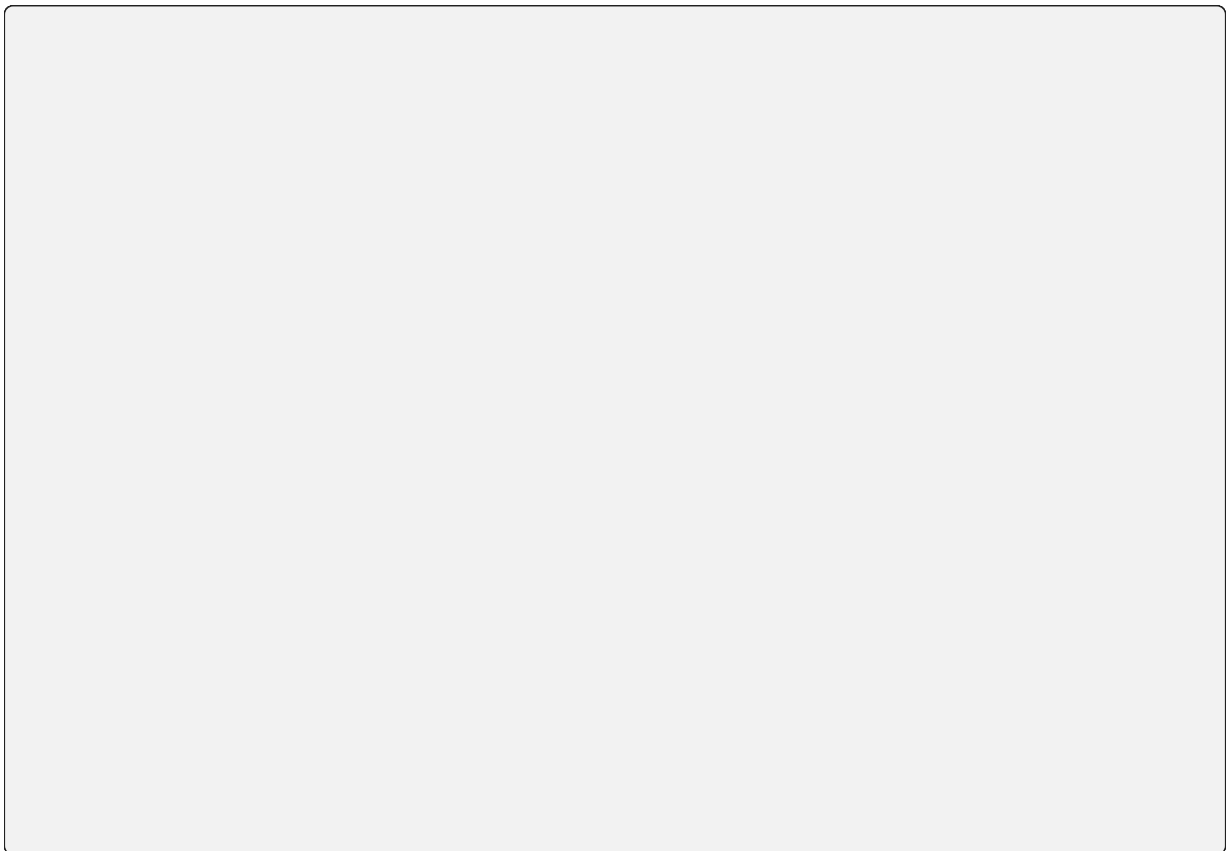
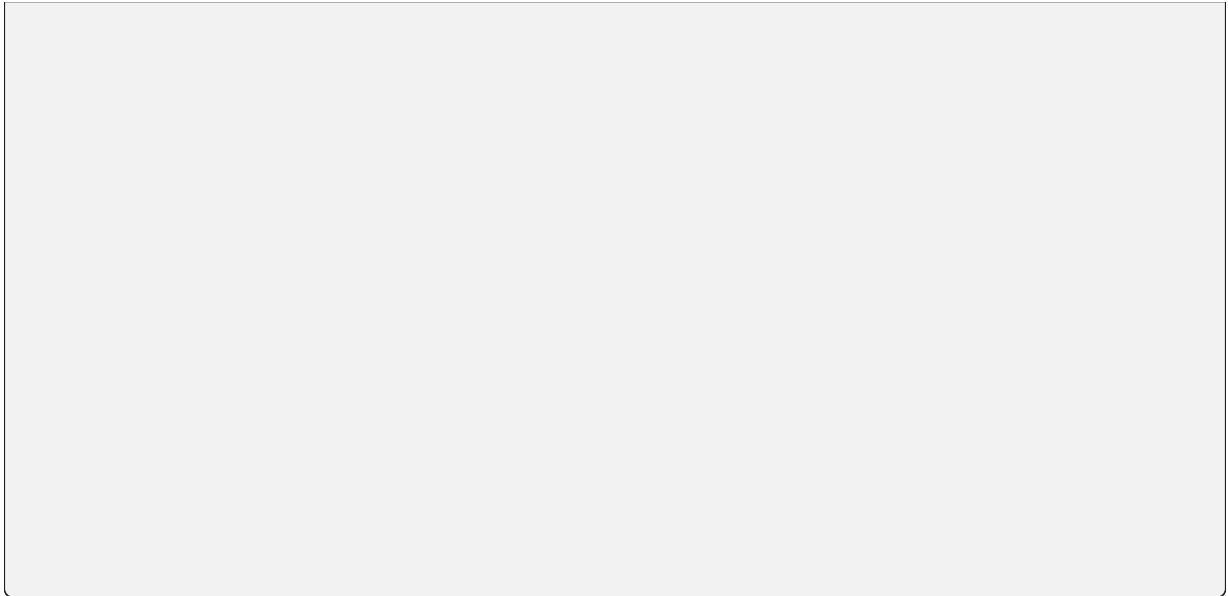
**Populate the database with initial fixtures**

**Sync the information of container registry**

,

**Run Backend.AI Manager service**

**Register systemd service**



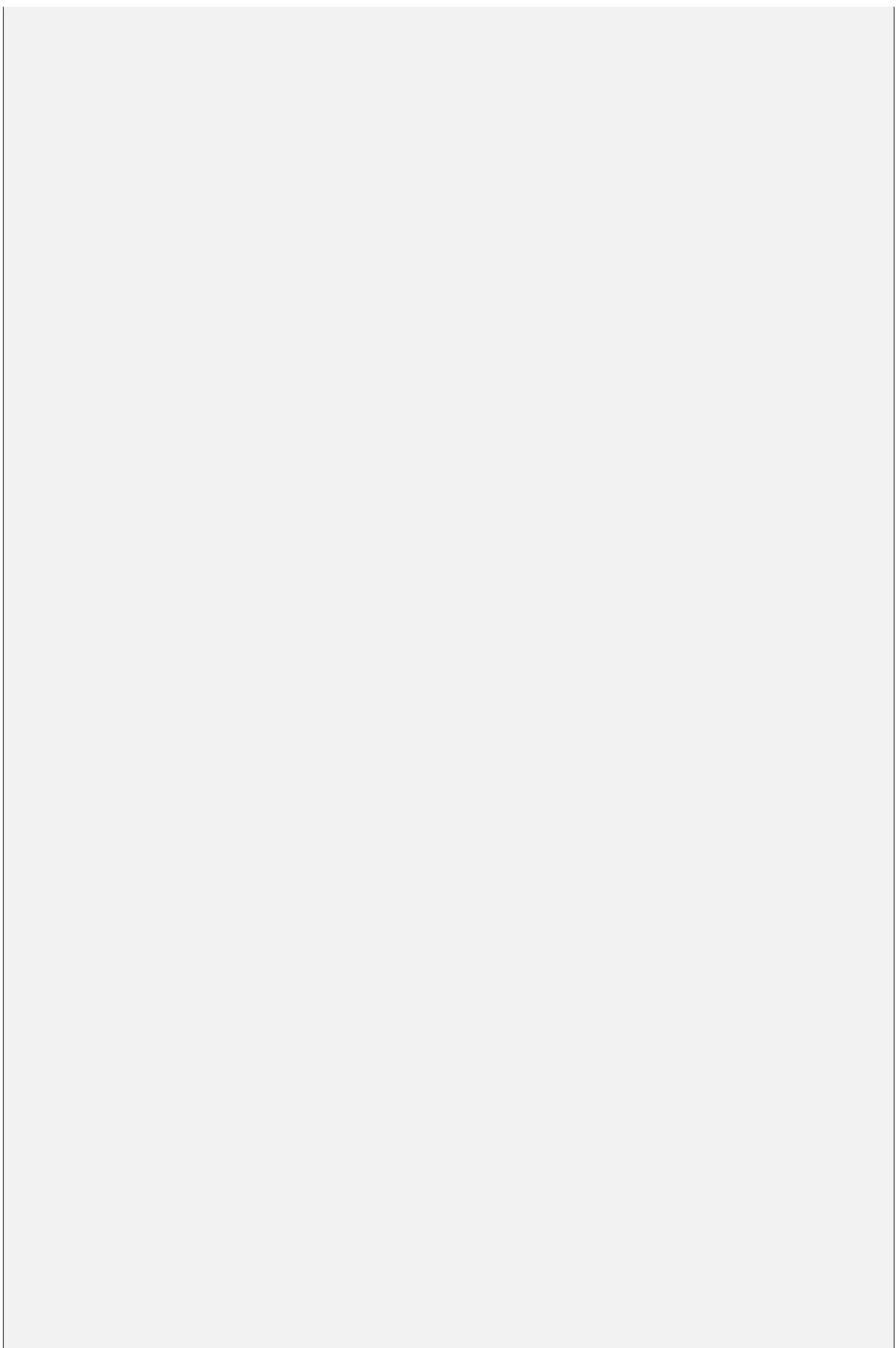
## 2.2.6 Install Backend.AI Agent

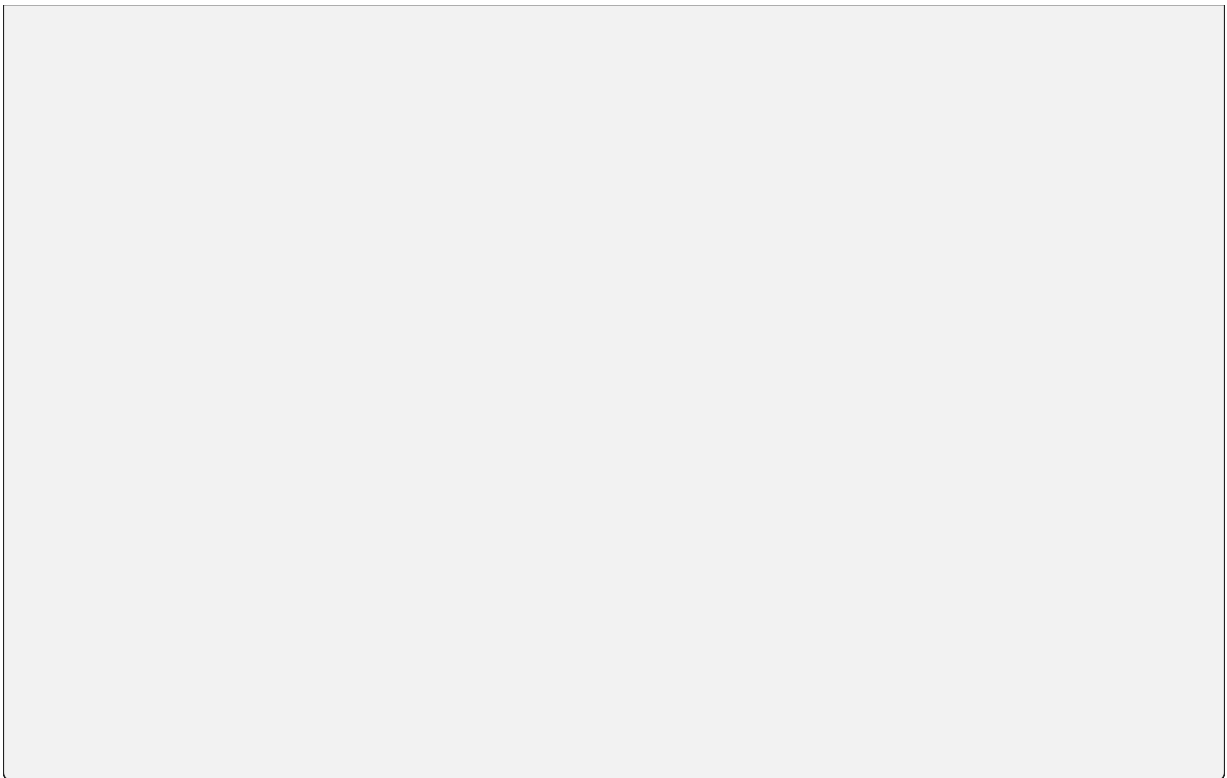
### Setting Up Accelerators

---

---

### Local configuration





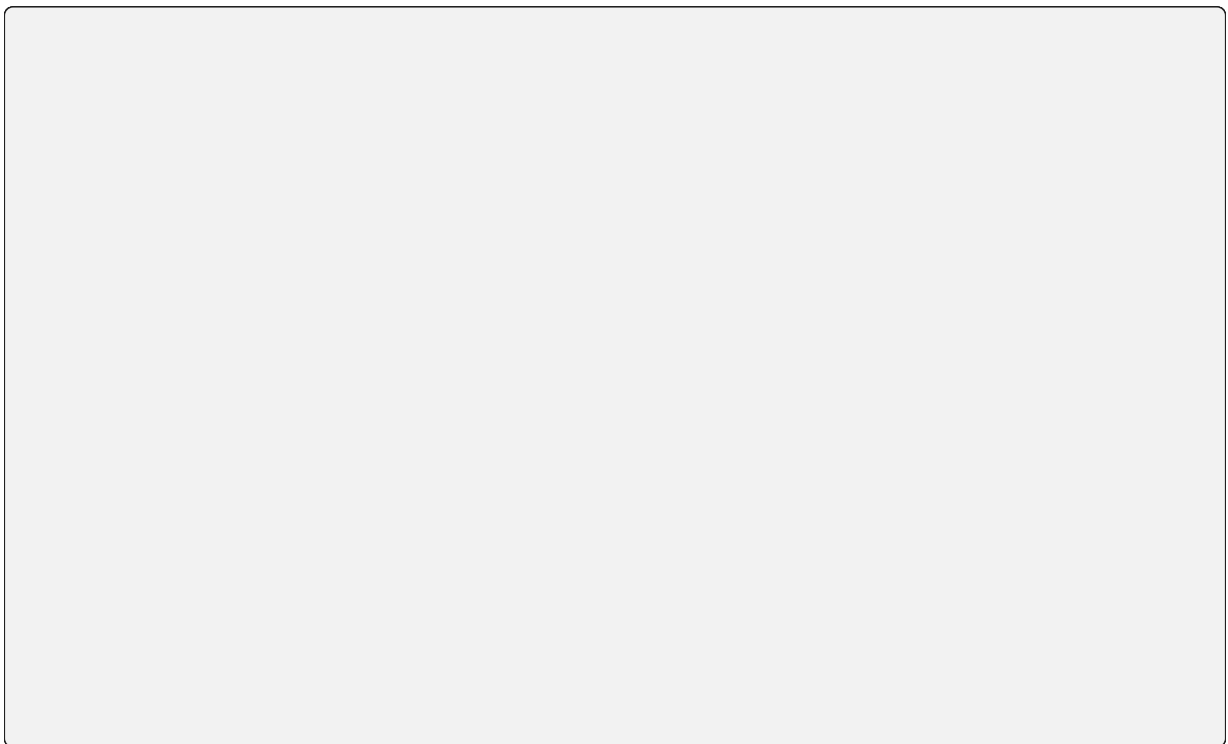
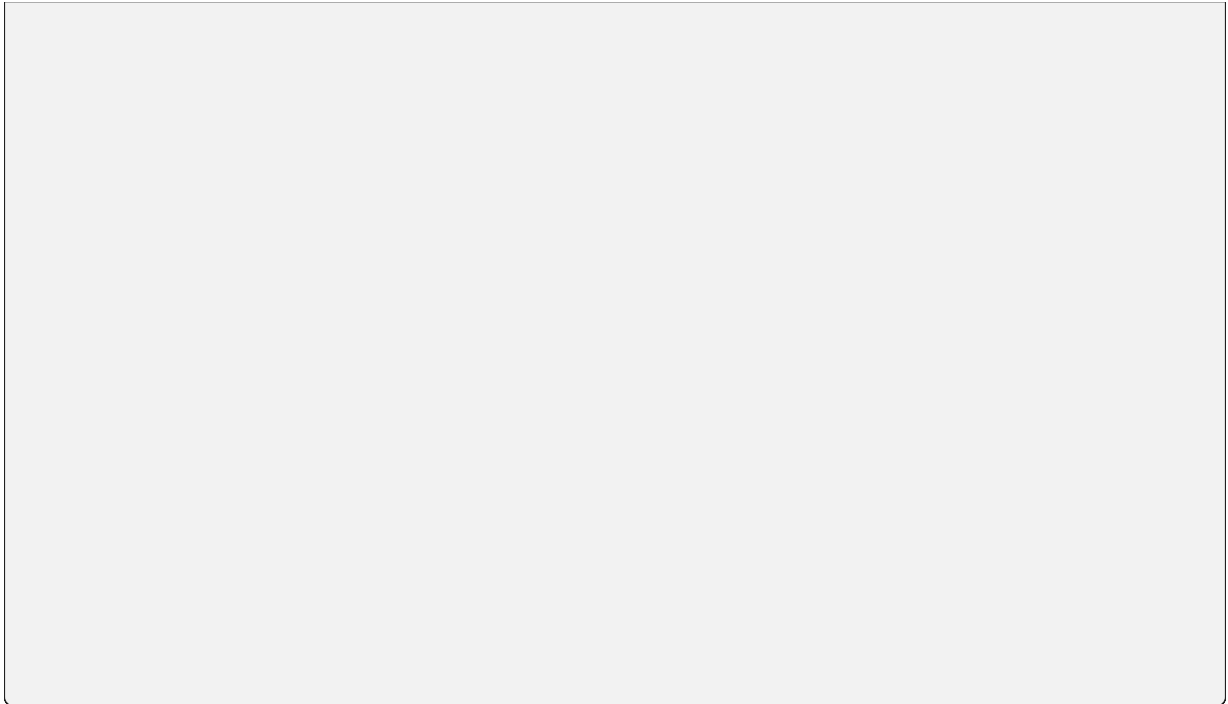
### Run Backend.AI Agent service

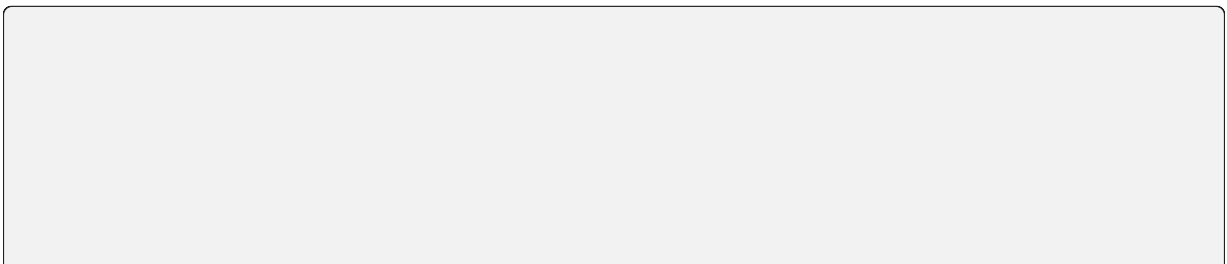
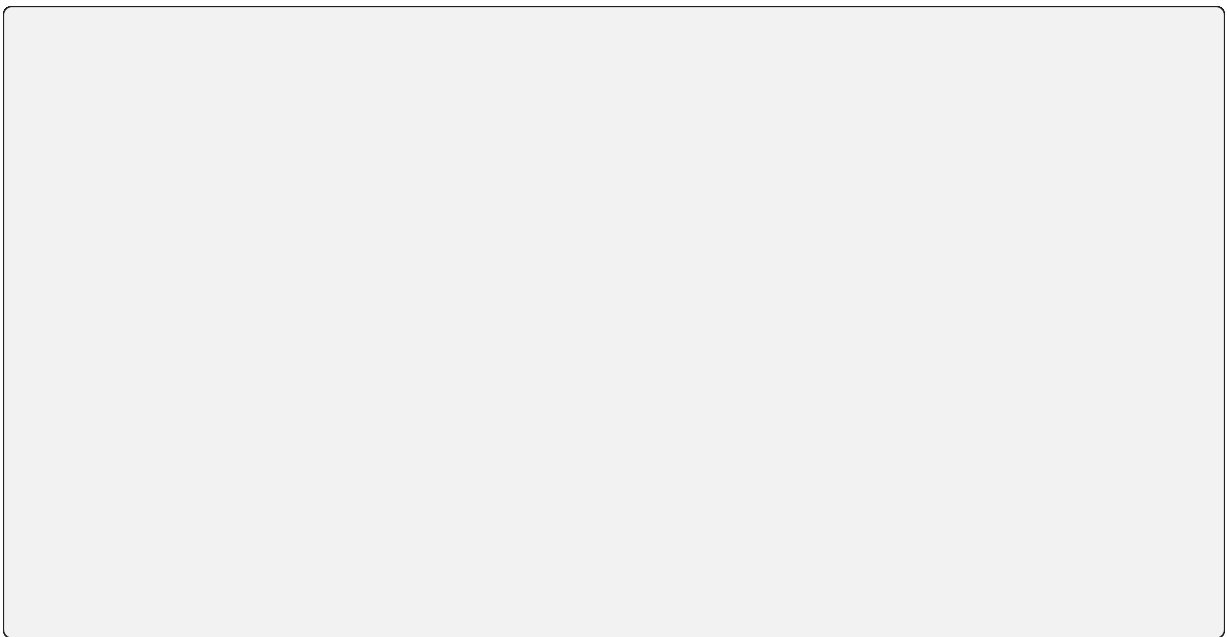
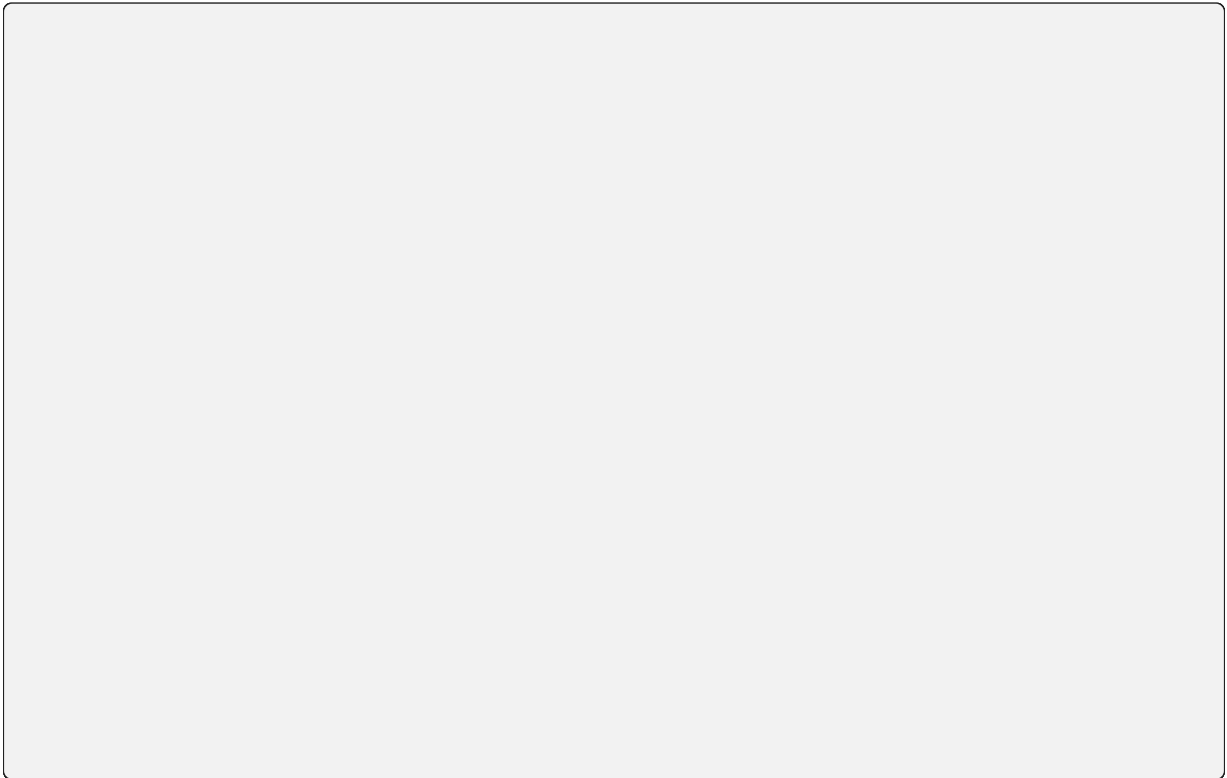


### Register systemd service







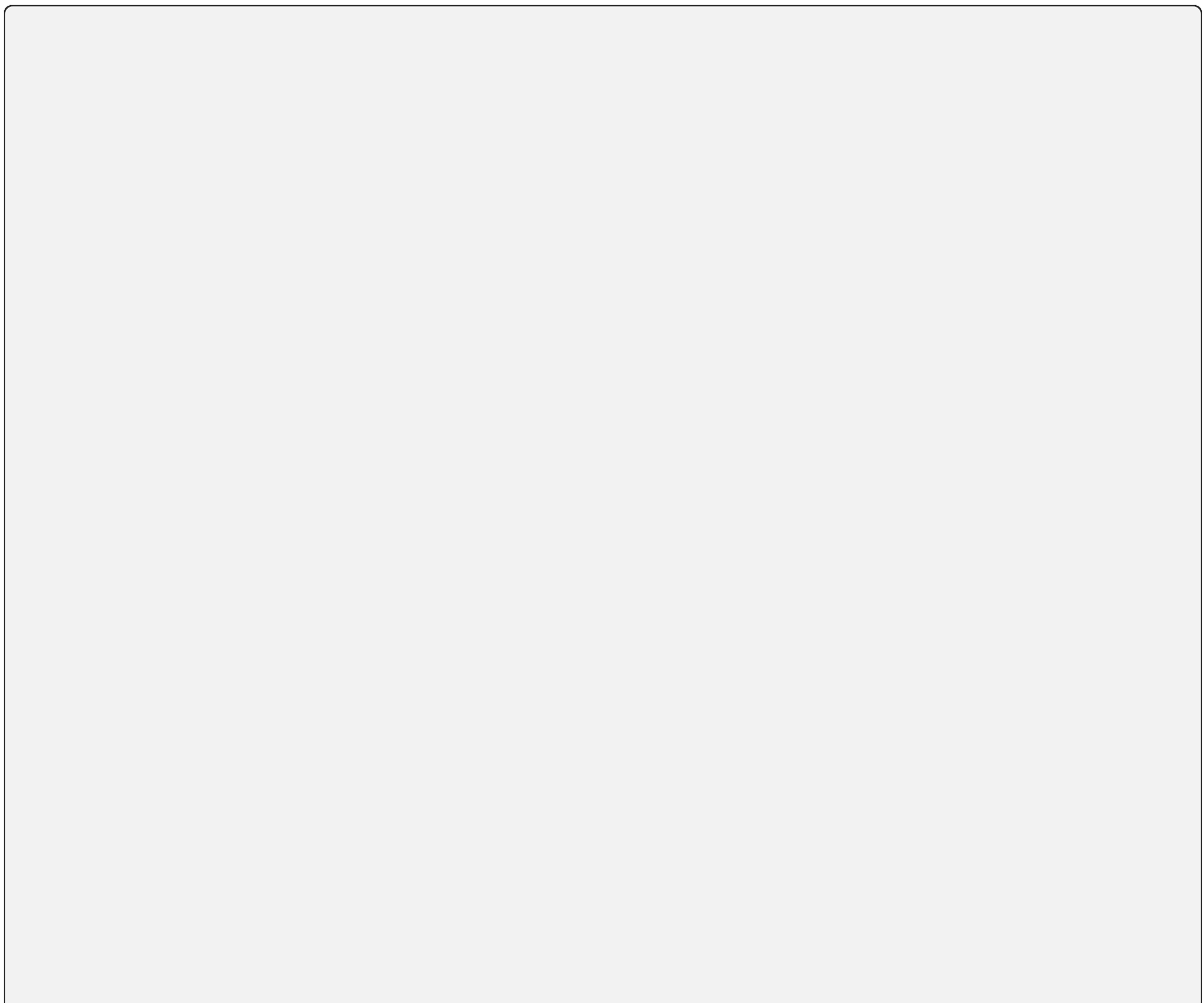


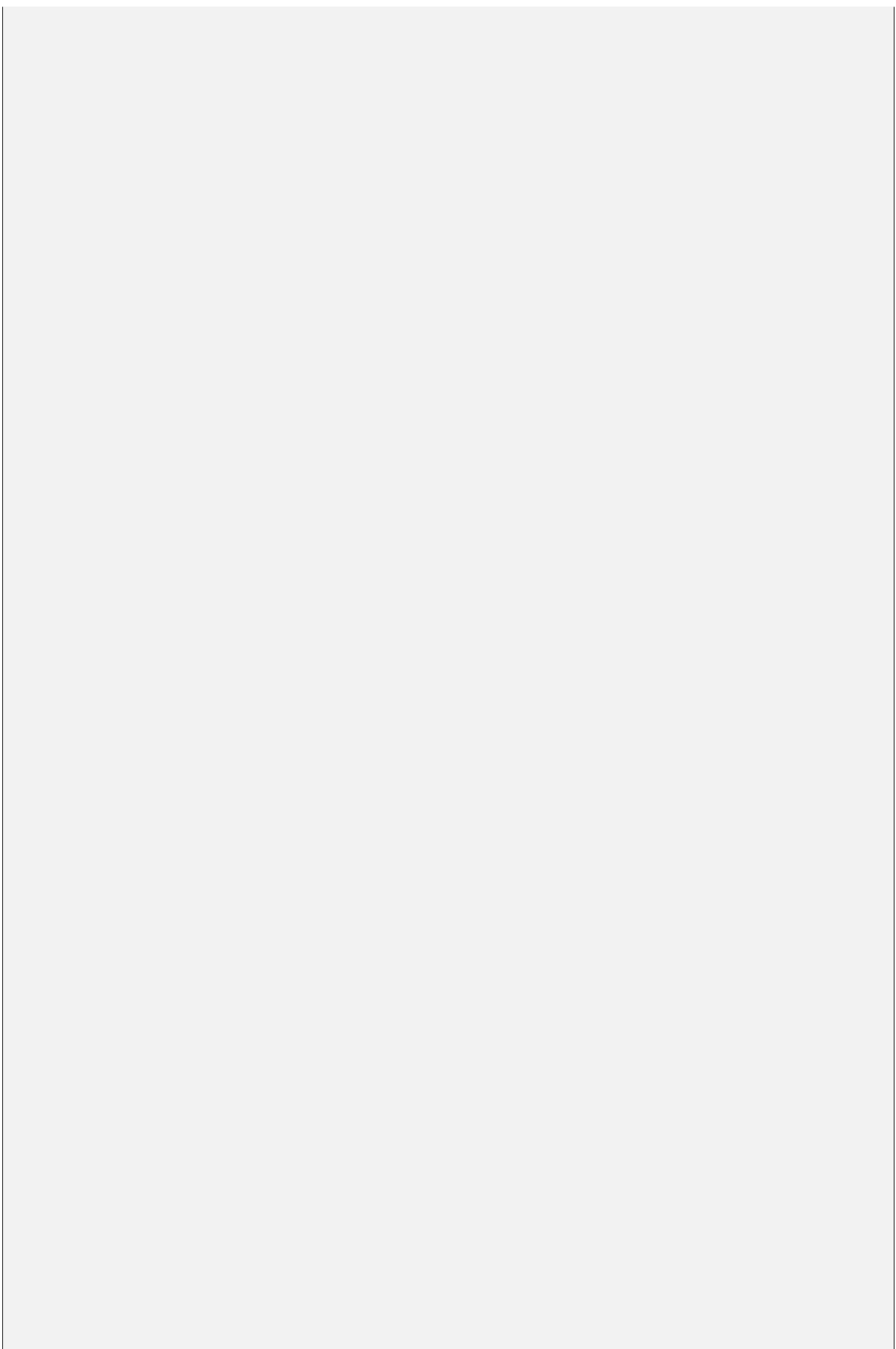


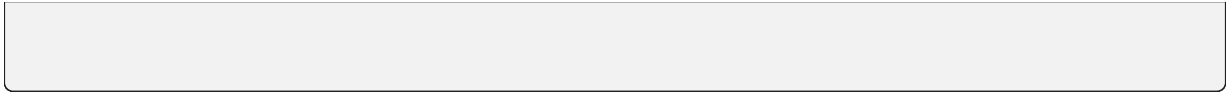
## 2.2.7 Install Backend.AI Storage Proxy



### Local configuration



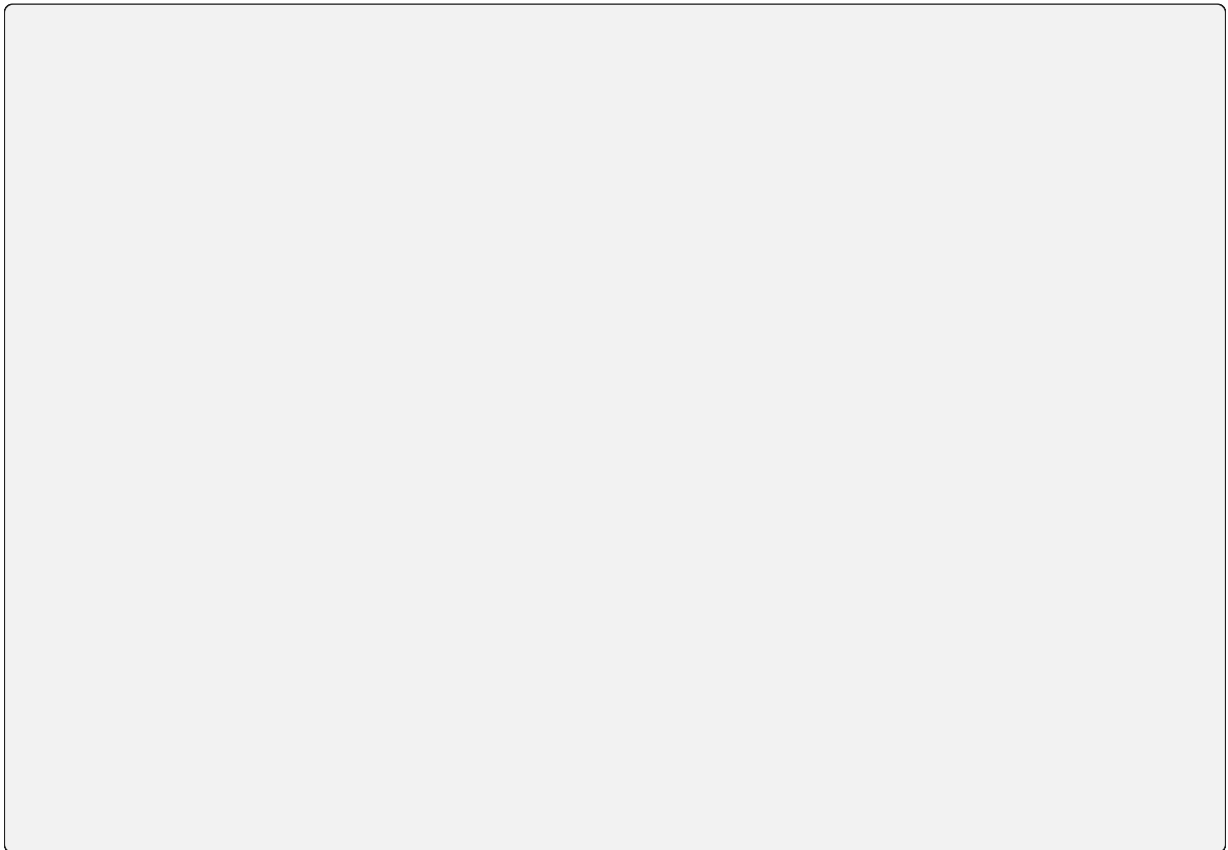


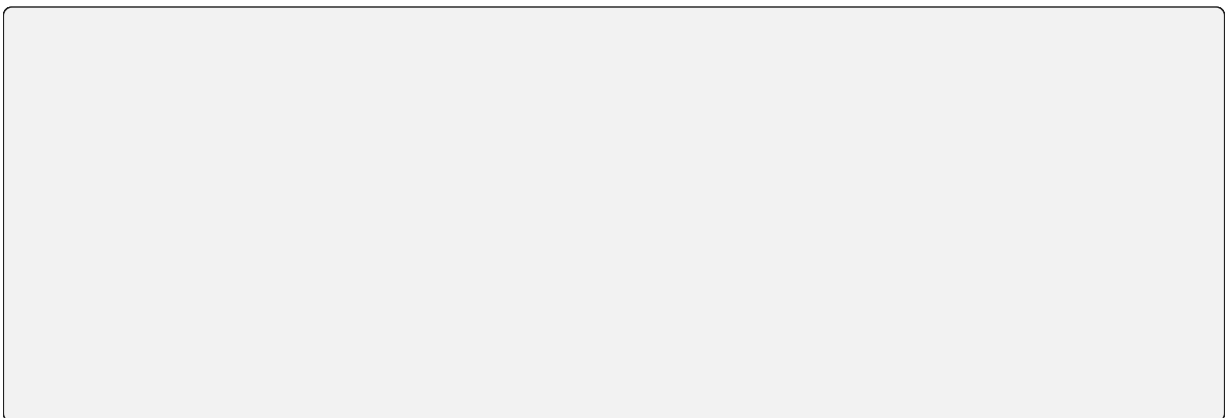
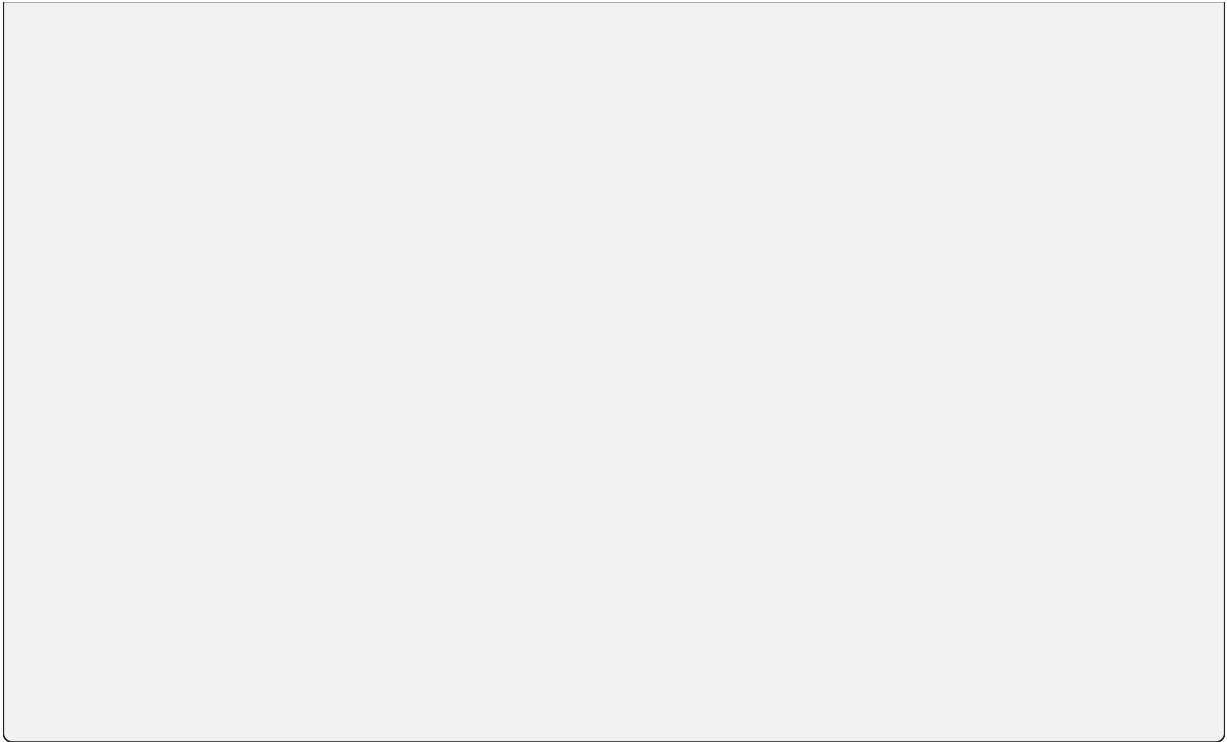


### Run Backend.AI Storage Proxy service



### Register systemd service

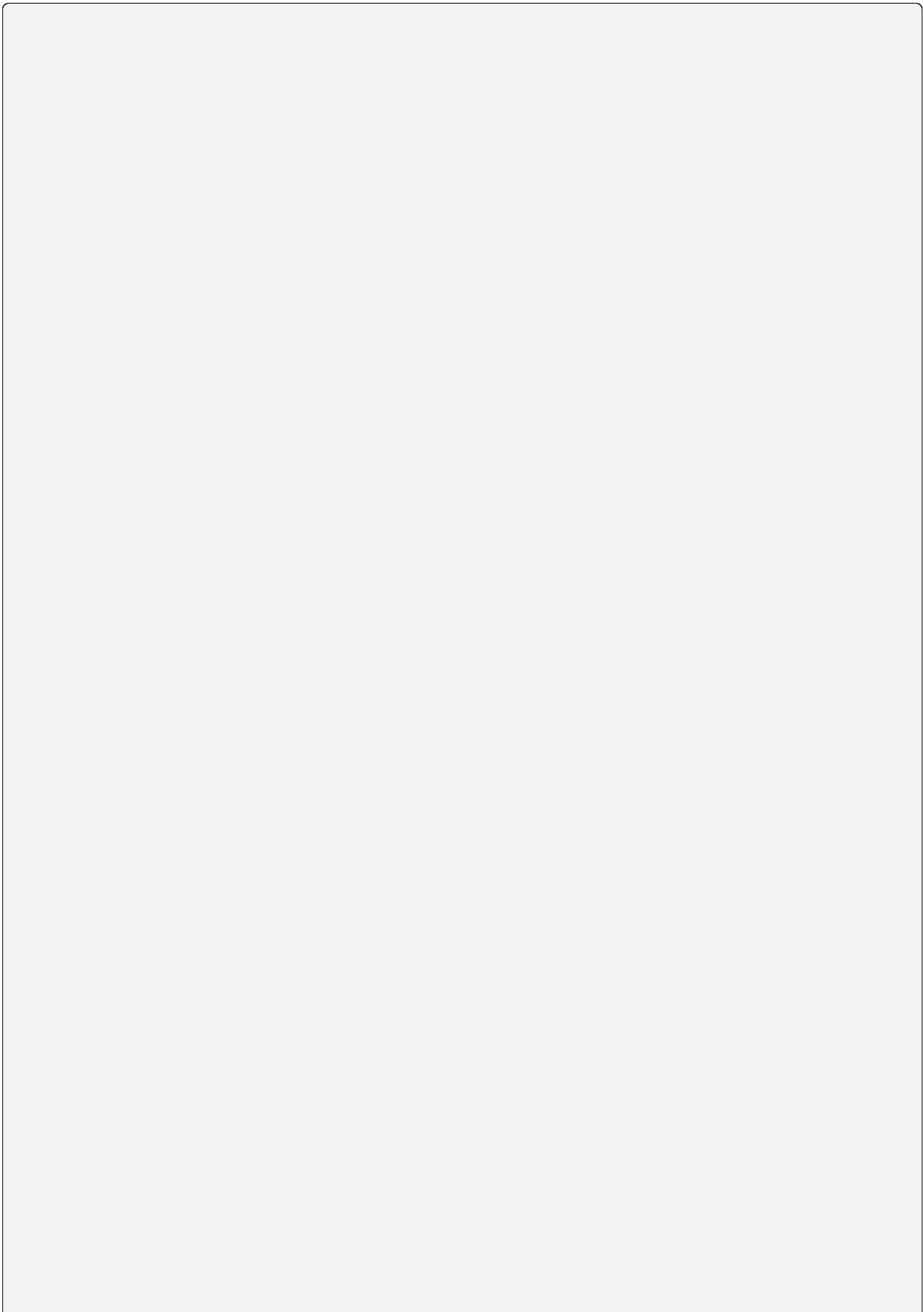


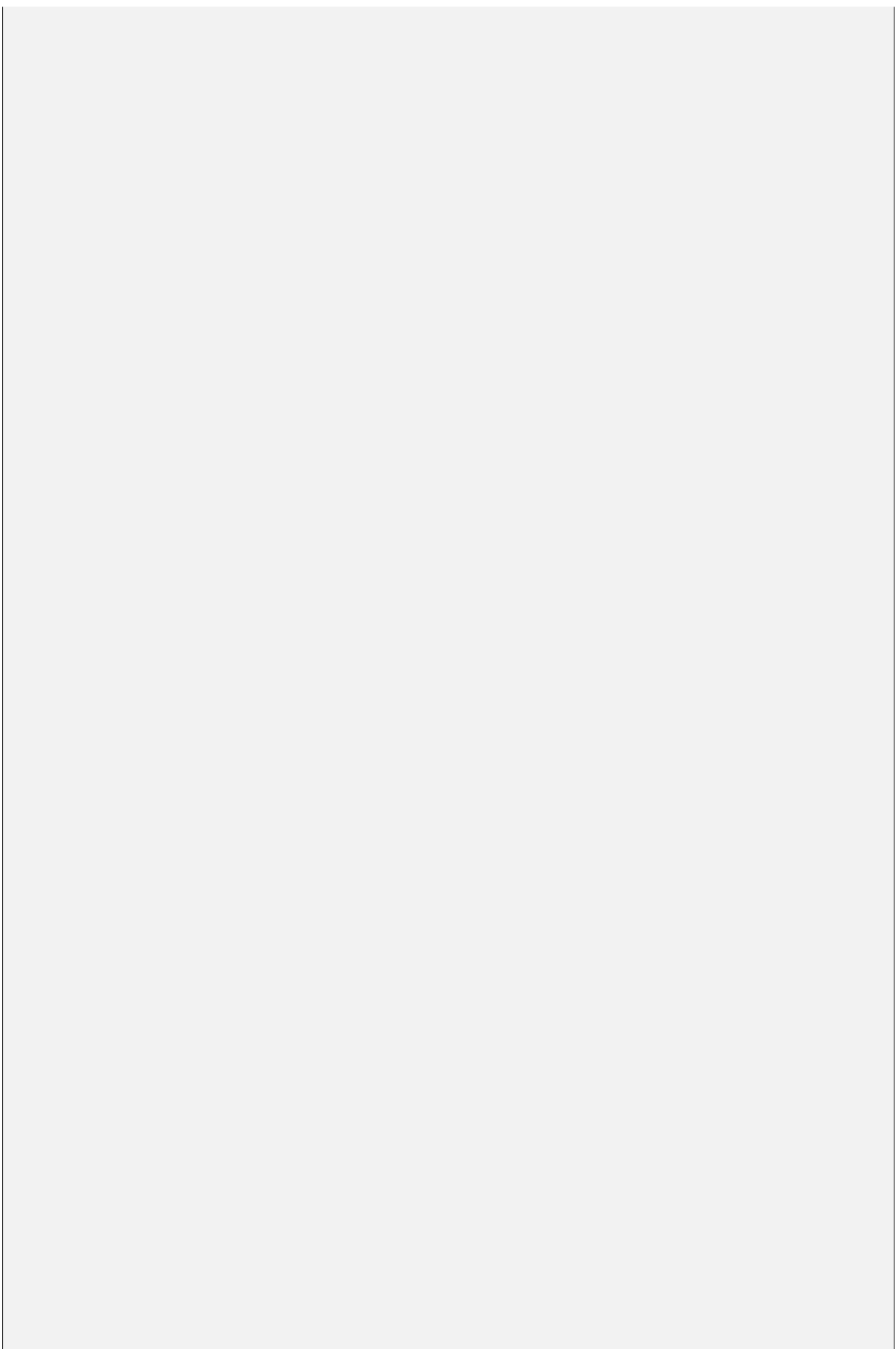


## 2.2.8 Install Backend.AI Webserver

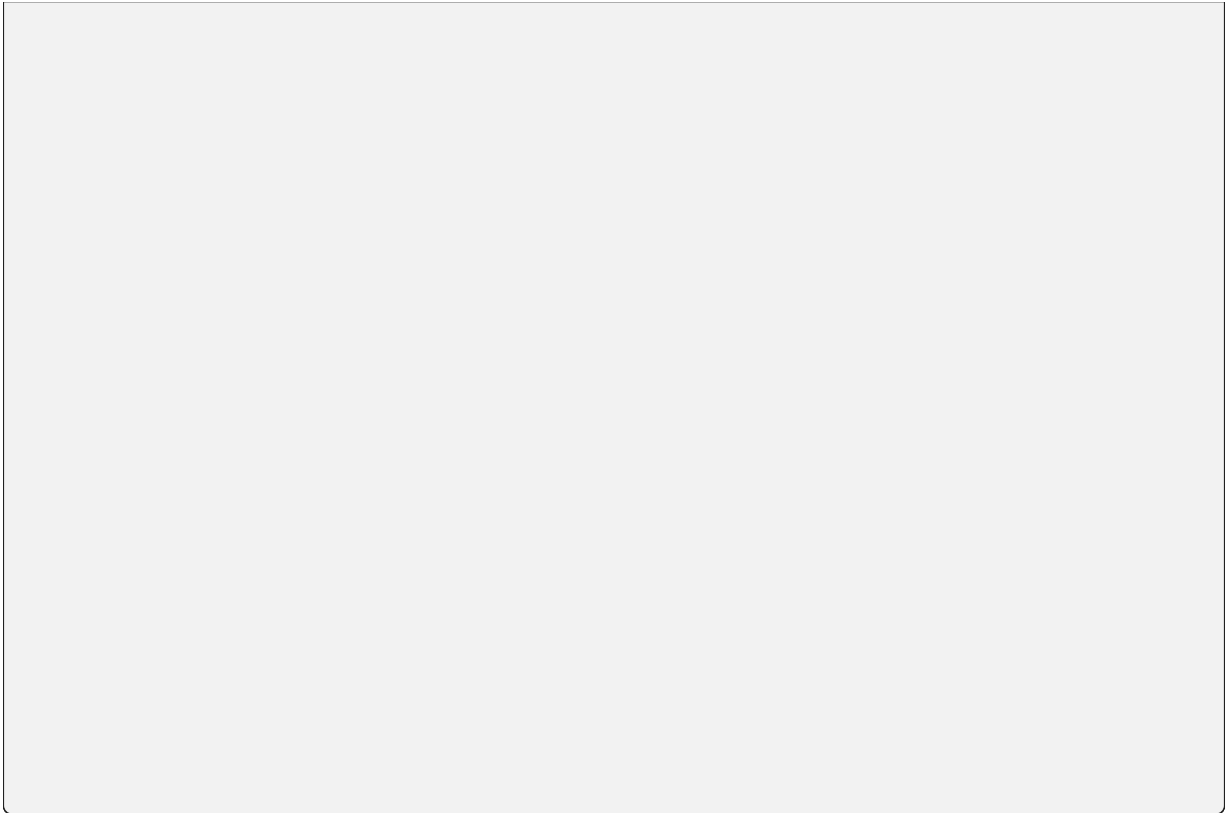


## Local configuration





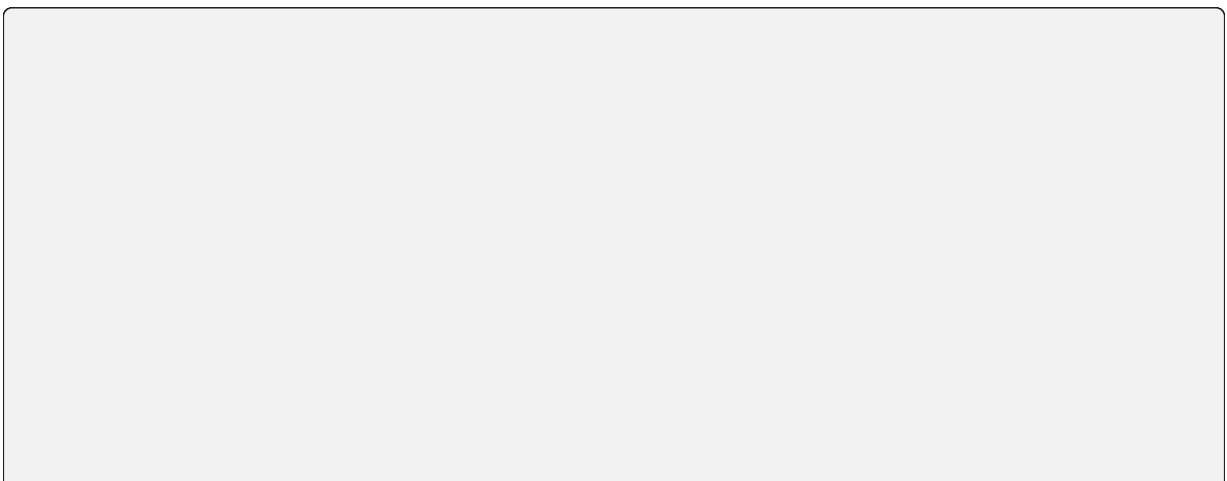


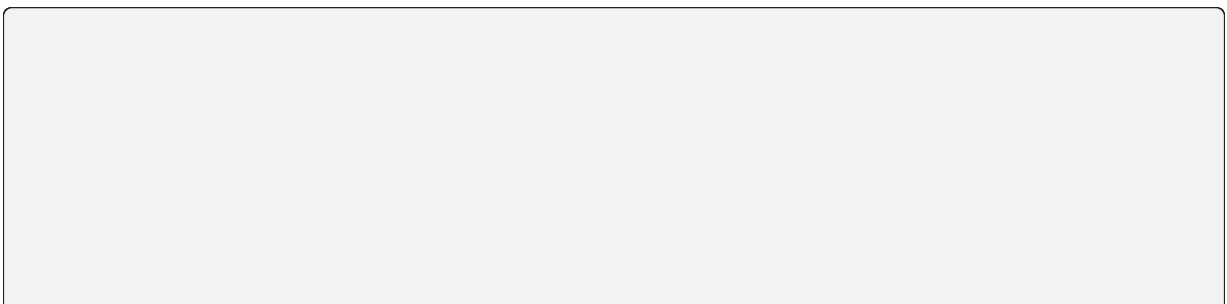
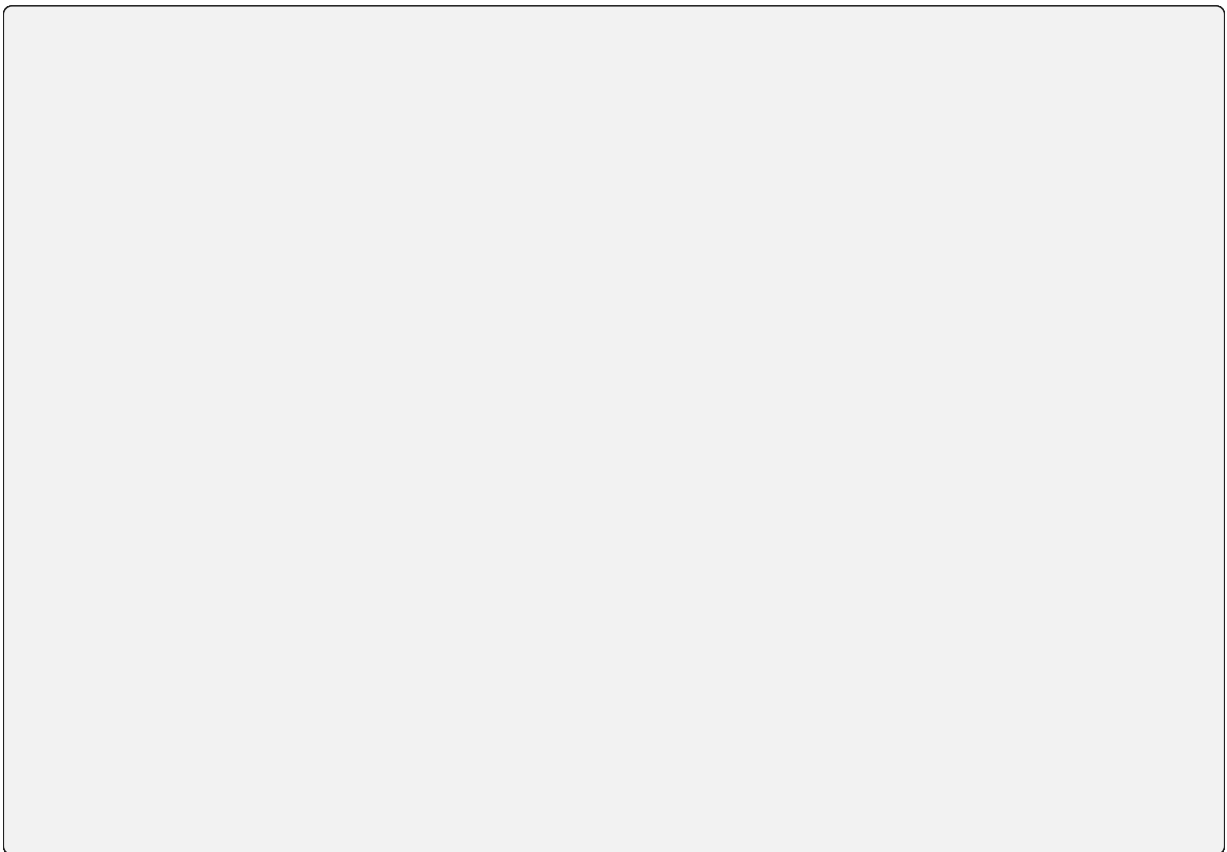
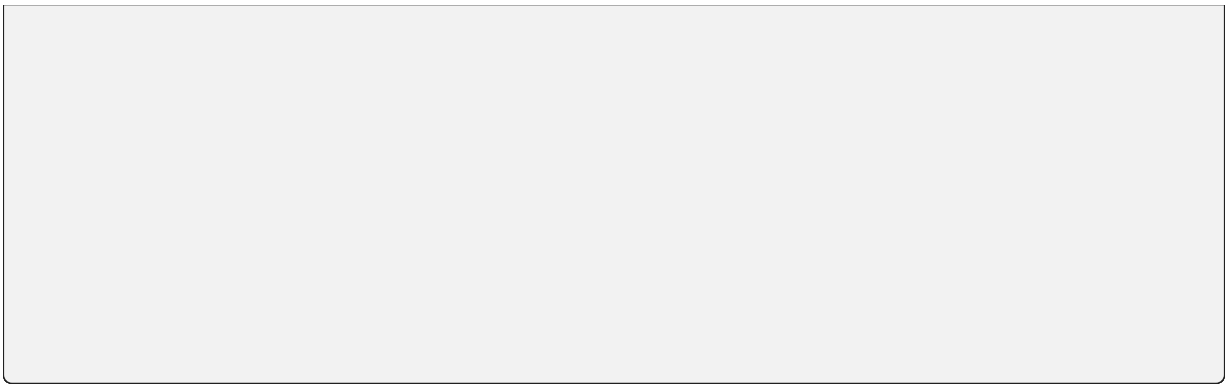


### Run Backend.AI Webserver service

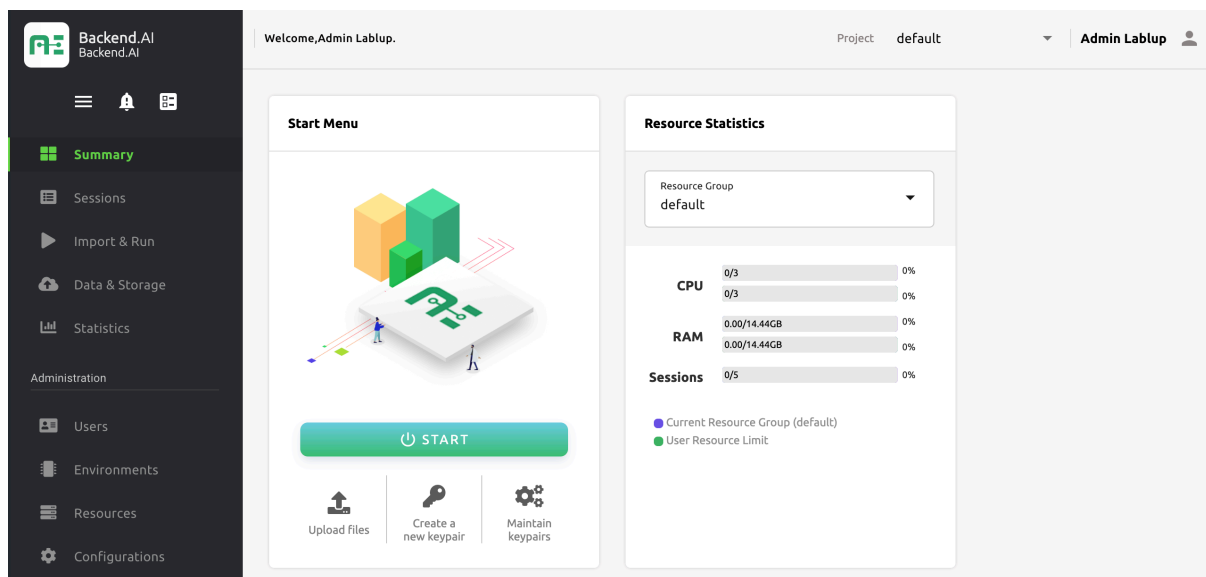
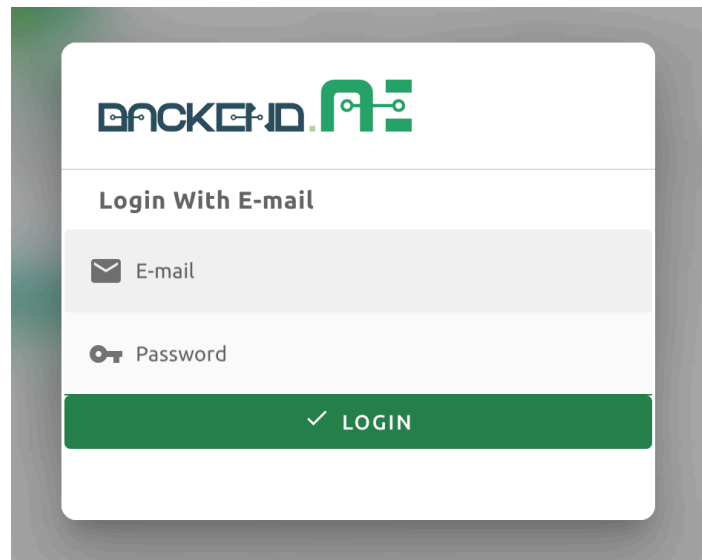


### Register systemd service

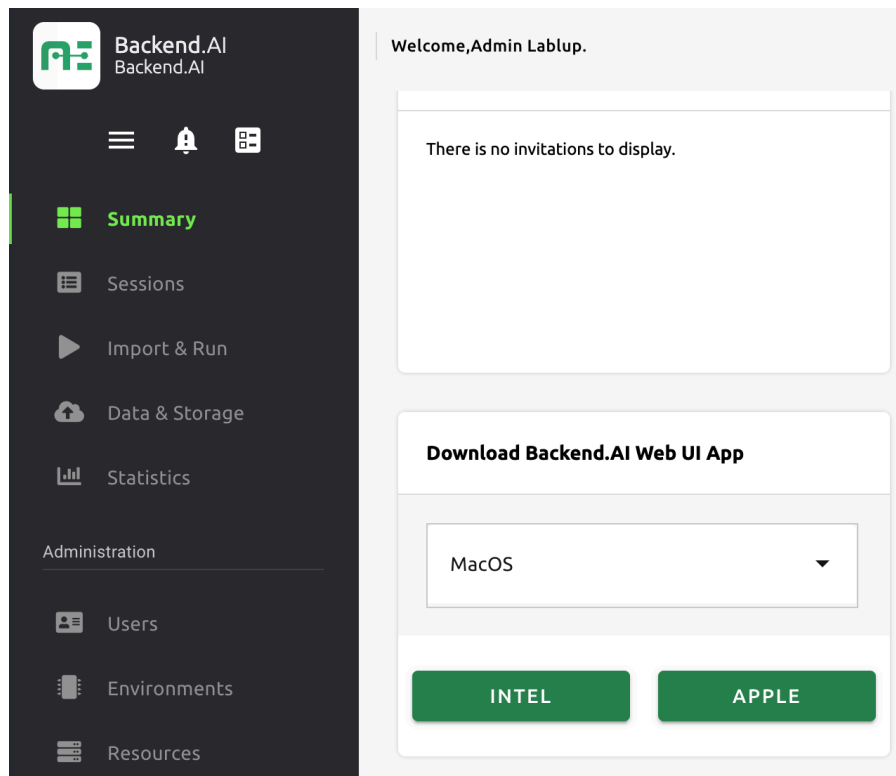




Check user GUI access via web



“”



## 2.3 Install on Clouds

## 2.4 Install on Premise

## 2.5 Install monitoring and logging tools

### 2.5.1 Guide variables

Name	Description

### 2.5.2 Install Datadog agent

### 2.5.3 Install Raven (Sentry client)

## 2.6 Upgrade existing Backend.AI cluster

---

---

---

---

### 2.6.1 Performing minor upgrade

,

Upgrading Backend.AI Manager

Upgrading other Backend.AI components

Others

### 2.6.2 Performing major upgrade

### Example of allowed upgrade paths

...

...

### Example of forbidden upgrade paths

...

### Upgrading Backend.AI Manager

### Upgrading other Backend.AI components

### Others

## 2.7 Environment specifics: WSL v2

### 2.7.1 Configuration of Docker Desktop for Windows

→ →

## 2.7.2 Configuration of WSL



## 2.7.3 Installation of Backend.AI



## 3.1 Install User Programs in Session Containers

### 3.1.1 Install packages with linuxbrew

Creating a user linuxbrew directory

,

Installing linuxbrew

,

## Testing linuxbrew

### Setting linuxbrew environment variables automatically

#### Example: Installing and testing htop

```
,  
,
```

```
,
```

## 3.1.2 Install packages with miniconda

### Creating a user miniconda-required directory

```
,
```

**miniconda test**

**Example: Installing and testing htop**

```
,
```

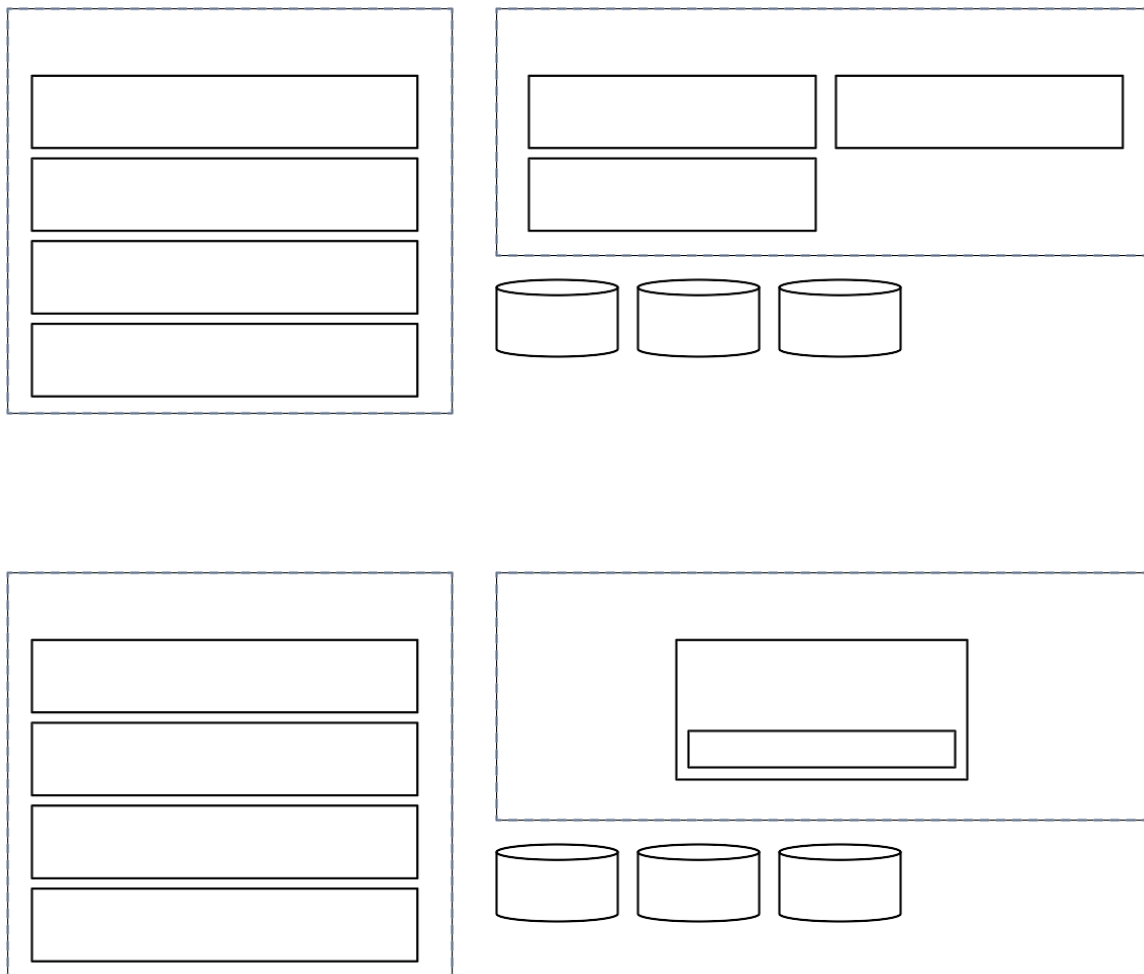
```
,
```

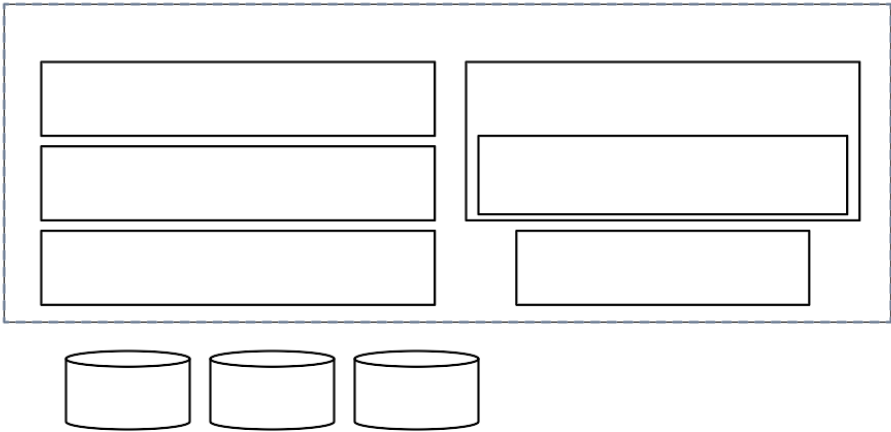
```
,
```



## DEVELOPER GUIDES

### 4.1 Development Setup





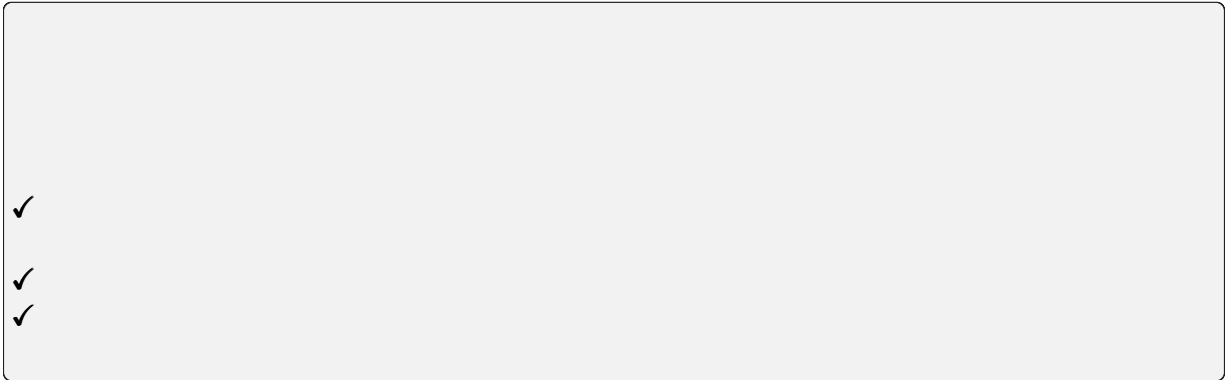
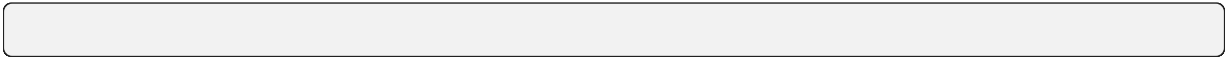
### 4.1.1 Installation from Source

#### Prerequisites

## Running the install-dev script

---

## Verifying Installation



Resetting the environment

,

Daily Workflows

## 4.2 Daily Development Workflows

### 4.2.1 About Pants

Key concepts





### Inspecting build configurations

---

### 4.2.2 Running lint and check

### 4.2.3 Running formatters

### 4.2.4 Running unit tests

### 4.2.5 Running integration tests

### 4.2.6 Building wheel packages

### 4.2.7 Using IDEs and editors

,

,

VSCode

,

Setting ID	Recommended value

Setting ID	Example value

---

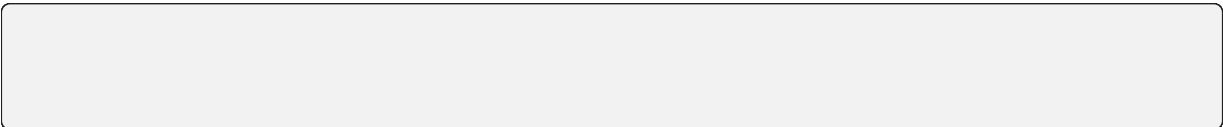
---

**Vim/NeoVim**

## Switching between branches

“”

### 4.2.8 Running entrypoints



### 4.2.9 Working with plugins



### 4.2.10 Writing test cases



```
“”
```

```
,
```

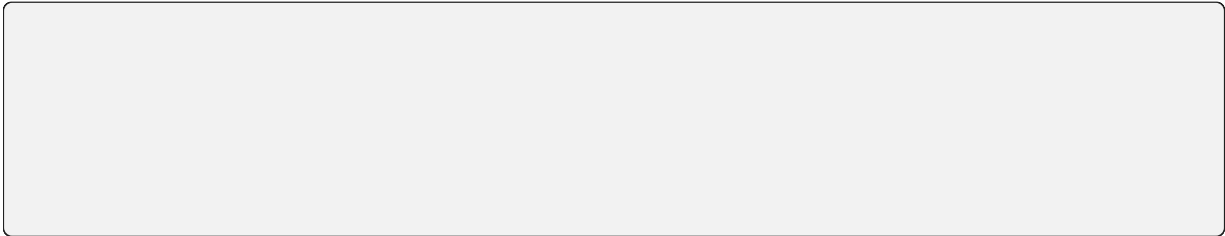
### 4.2.11 Writing documentation

```
,
```

### 4.2.12 Advanced Topics

#### Adding new external dependencies

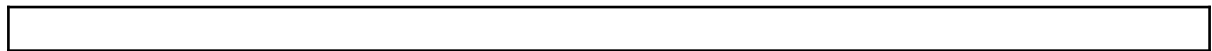
## Merging lockfile conflicts



## Resetting Pants

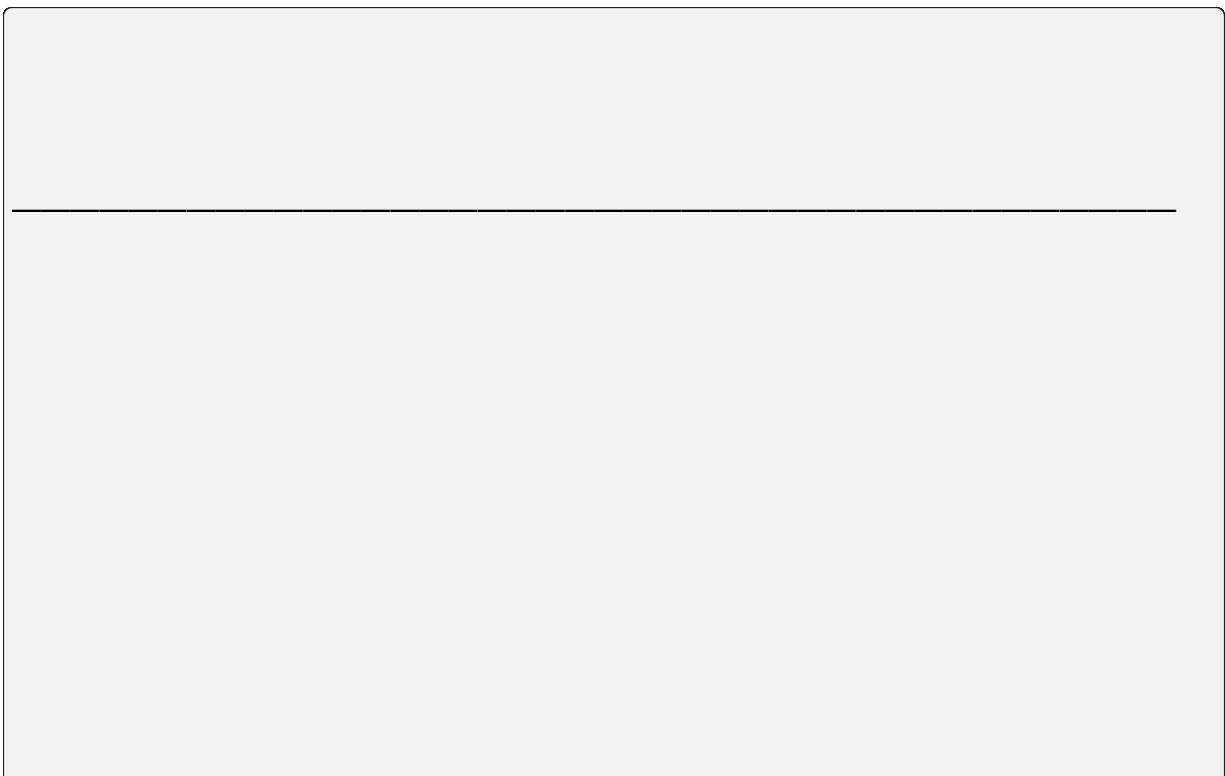


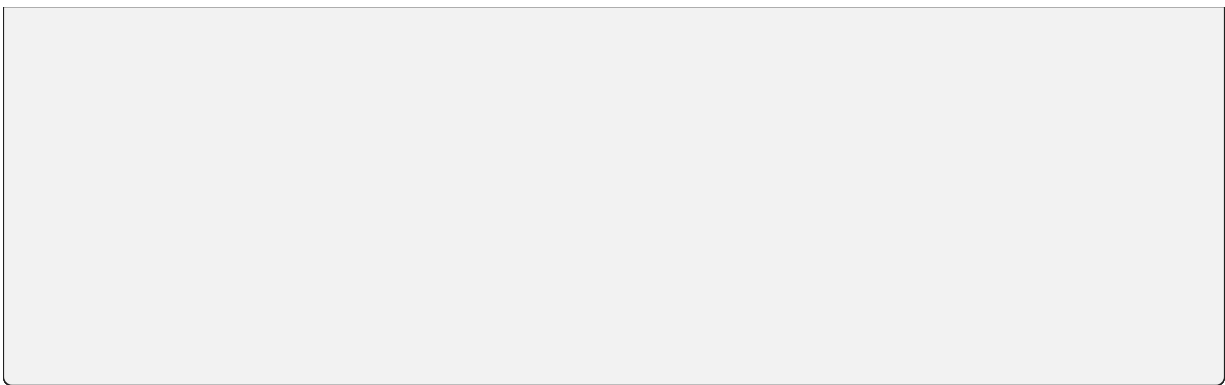
,



Resolve the error message ‘Pants is not available for your platform’, When installing Backend.AI with pants

“



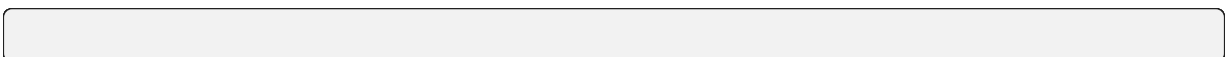


### Resolving missing directories error when running Pants



### Changing or updating the Python runtime for Pants

### Debugging test cases (or interactively running test cases)



### Installing a subset of mono-repo packages in the editable mode for other projects

,







,

### Boosting the performance of Pants commands



### Making a new release

“”

.

“” “”

,



Backporting to legacy per-pkg repositories

## 4.3 Adding New Kernel Images

### 4.3.1 Overview

### 4.3.2 Kernel Runner

“”

”“

#### Image Prerequisites

,

### 4.3.3 Metadata Labels

“” “” “”

,

### 4.3.4 Service Ports

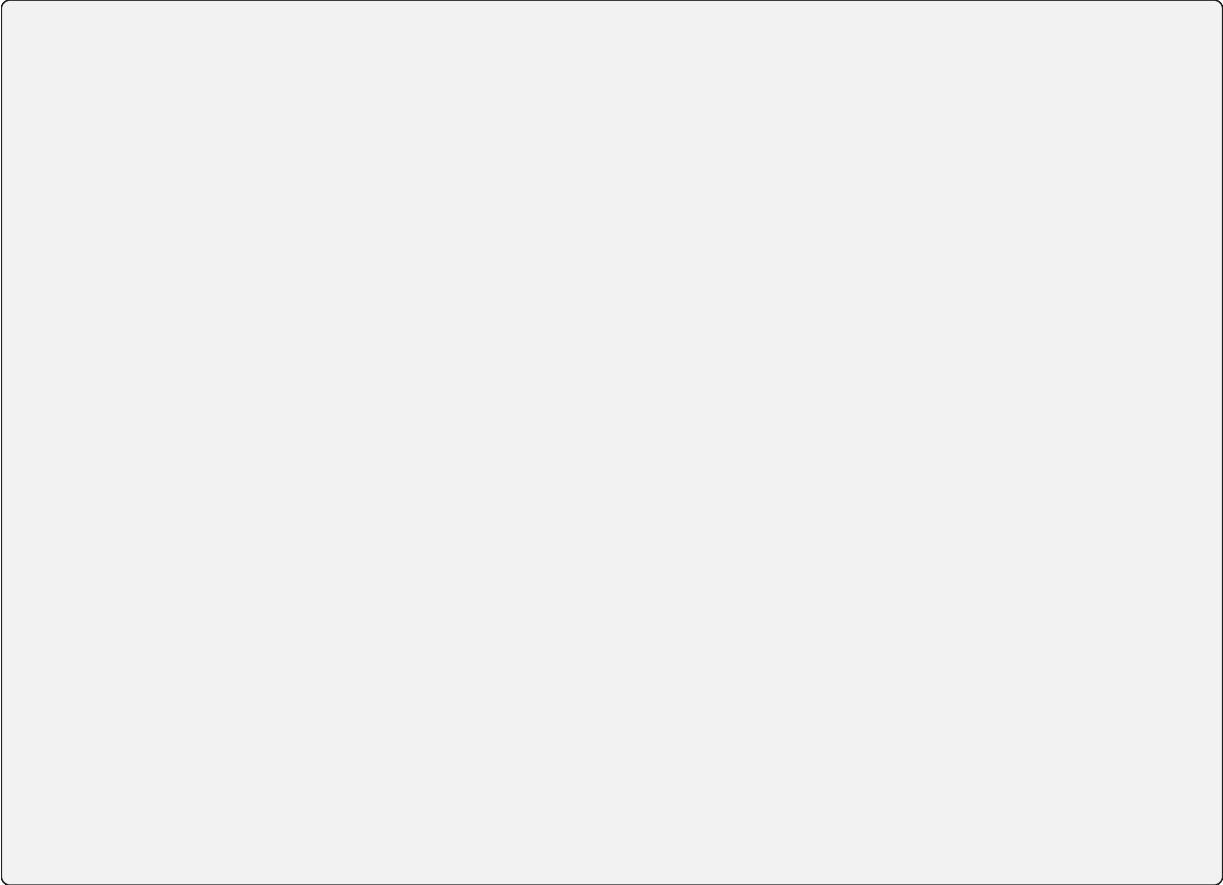
“”

#### Port Mapping Declaration

## Service Definition DSL

,

“”



“” ,

## Available predefined variables

Available prestart actions

“”.

Action Name	Arguments	Return
	:	
	“”	
	:	
	“”	
	:	
	:	
	:	

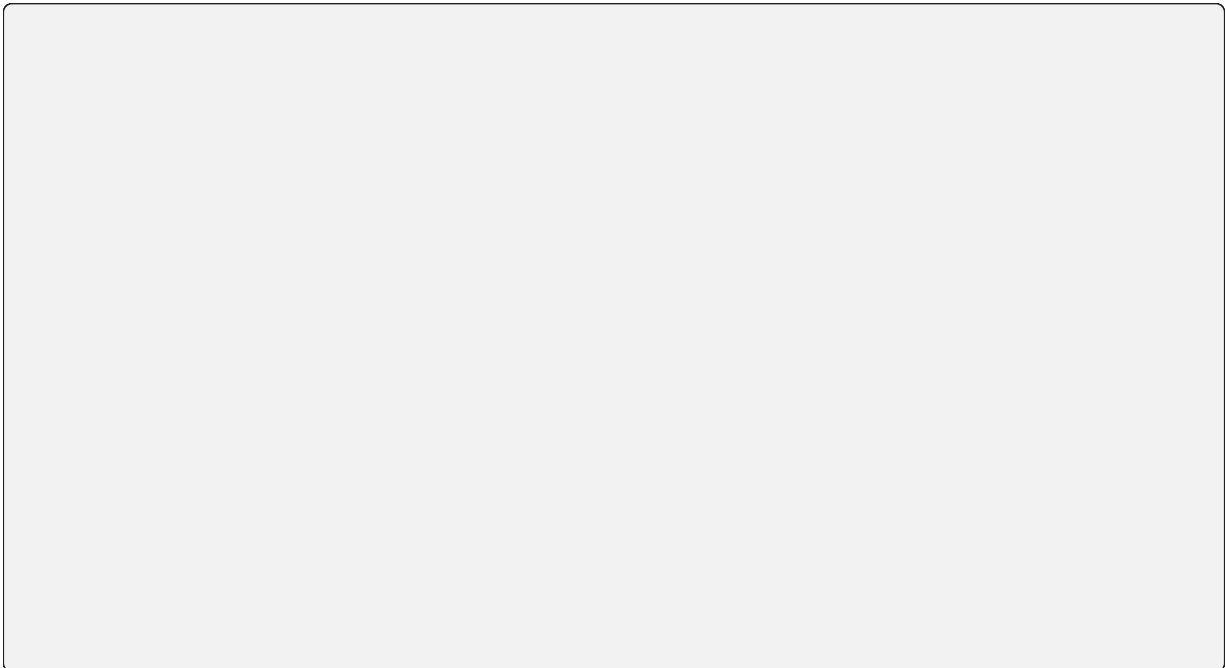
Interpretation of URL template

,

### 4.3.5 Jail Policy

Adding Custom Jail Policy

### 4.3.6 Example: An Ubuntu-based Kernel



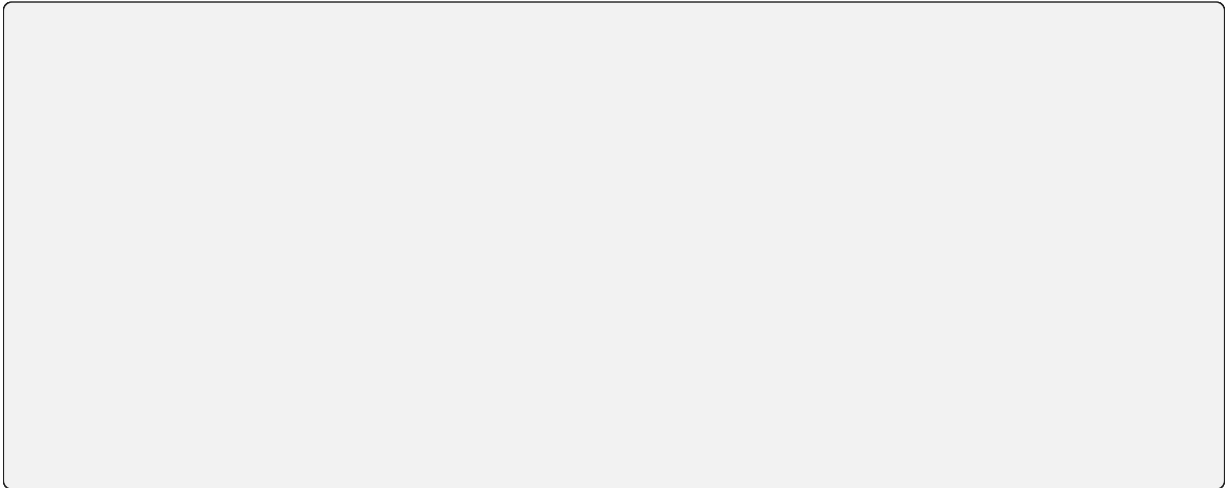
### 4.3.7 Custom startup scripts (aka custom entrypoint)

### 4.3.8 Implementation details

#### The query mode I/O protocol

,

,



#### The pseudo-terminal mode protocol

“”

“” “”

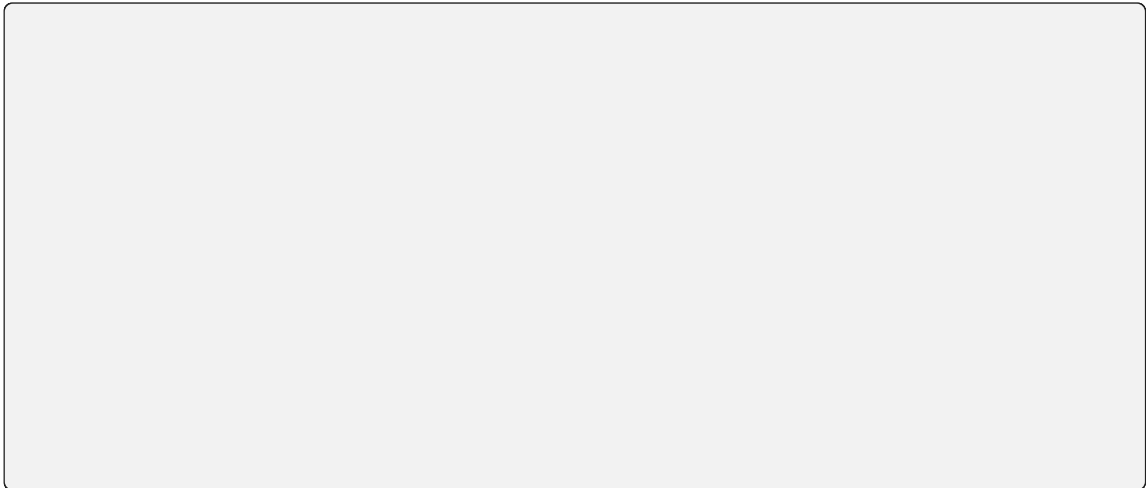
,

“”

## 4.4 Using Mocked Accelerators

### 4.4.1 Configuring the mock-accelerator plugin

,





#### 4.4.2 Activating the mock-accelerator plugin

,

### 4.5 Version Numbering

,

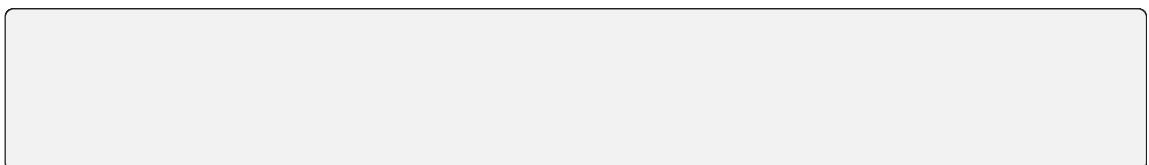
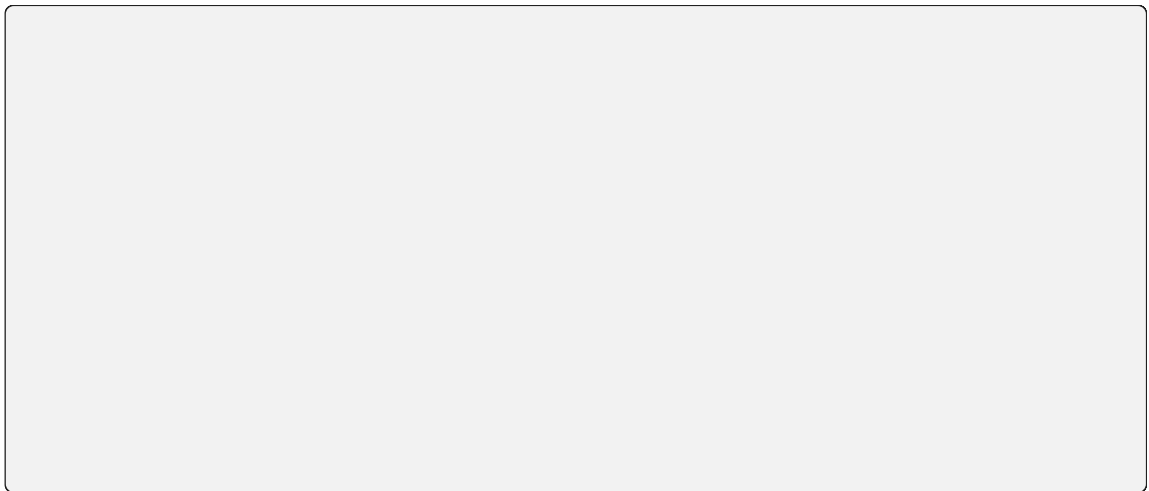
### 4.6 Upgrading



## MIGRATION GUIDES

### 5.1 Upgrading from 20.03 to 20.09

### 5.2 Migrating from the Docker Hub to cr.backend.ai



“” “...”



## BACKEND.AI MANAGER REFERENCE

### 6.1 Manager API Common Concepts

#### 6.1.1 API and Document Conventions

##### HTTP Methods

##### Parameters in URI and JSON Request Body

##### HTTP Status Codes and JSON Response Body

##### JSON Field Notation

Example	Meaning

JSON Value Types

,

Type	Description

API Versioning



6.1.2 JSON Object References

Paging Query Object

Key	Type	Description

Paging Info Object

Key	Type	Description

## KeyPair Item Object

Key	Type	Description

## KeyPair Properties Object

Key	Type	Description

Key	Type	Description

## Service Port Object

[illegible]

## Batch Execution Query Object

--

## Execution Result Object

## Console Item Object

Key	Type	Description
-----	------	-------------



Execution Result File Object

Key	Typ	Description

Container Stats Object

Key	Typ	Description

Creation Config Object

Key	Typ	Description
		,

## Resource Slot Object

Key	Type	Description
		<p>“” , “” , “” , “” , “” , “” ,</p> <div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 10px auto;"> <math>2^{53} - 1</math> </div>
		<p>“” “”</p> <p>“”</p> <p>“”</p>

## Resource Preset Object

Key	Type	Description
	" , " , "	

## Virtual Folder Creation Result Object

Key	Type	Description

## Virtual Folder List Item Object

Key	Type	Description
		"", "", "" "" ""
		, ,
		"", "

Virtual Folder Item Object

Key	Type	Description
		,
		“” “” .

Virtual Folder File Object

Key	Type	Description
		,
		,

Virtual Folder Invitation Object

Key	Type	Description
		,

Key	Type	Description
		“” “
		,

### 6.1.3 Authentication

#### Access Tokens and Secret Key

#### Common Structure of API Requests

HTTP Headers	Values
	,

#### Common Structure of API Responses

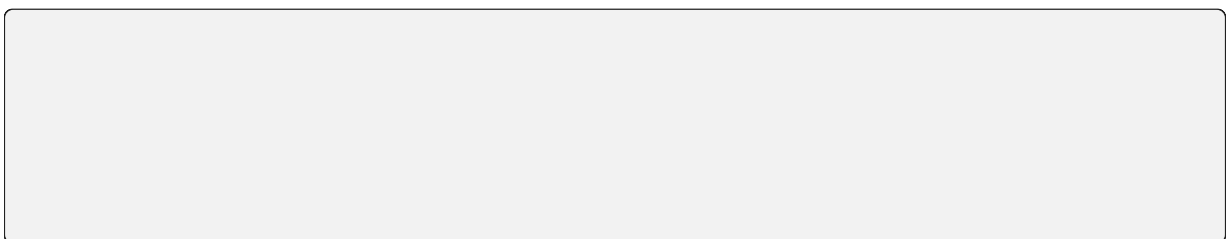
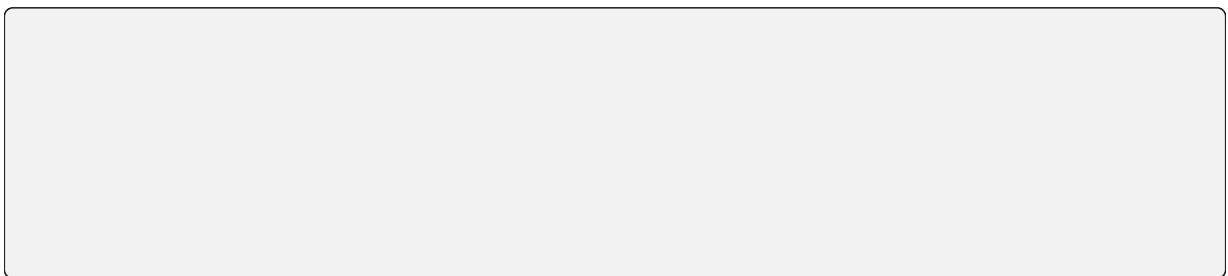
HTTP Headers	Values

#### Signing API Requests

#### Generating a signing key



**Generating a string to sign**

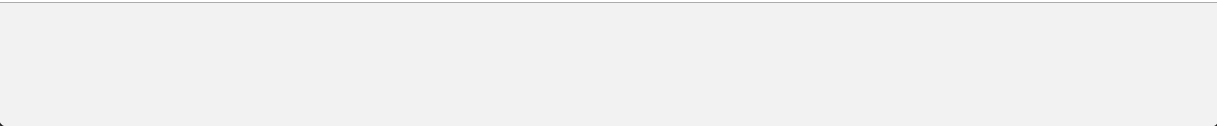


### Attaching the signature

### Example Requests and Responses

#### Success example for checking the latest API version

#### Failure example with a missing authorization header



### 6.1.4 Rate Limiting

,

HTTP Headers	Values

“”.

## 6.2 Manager REST API

### 6.2.1 Session Management

#### Creating Session

## Parameters





Response

HTTP Status Code	Description

Key	Type	Description

Destroying Session

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

Key	Type	Description

Restarting Session

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

6.2.2 Code Execution (Query Mode)

Executing Snippet

Parameters

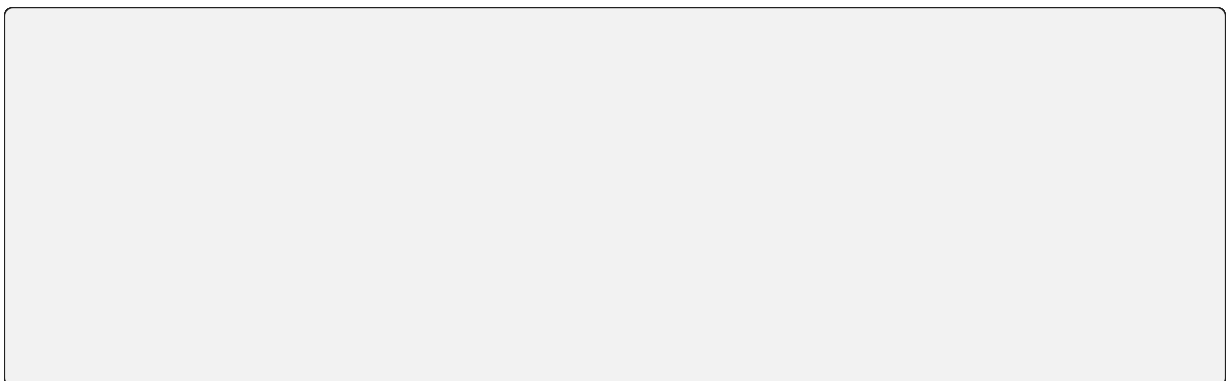
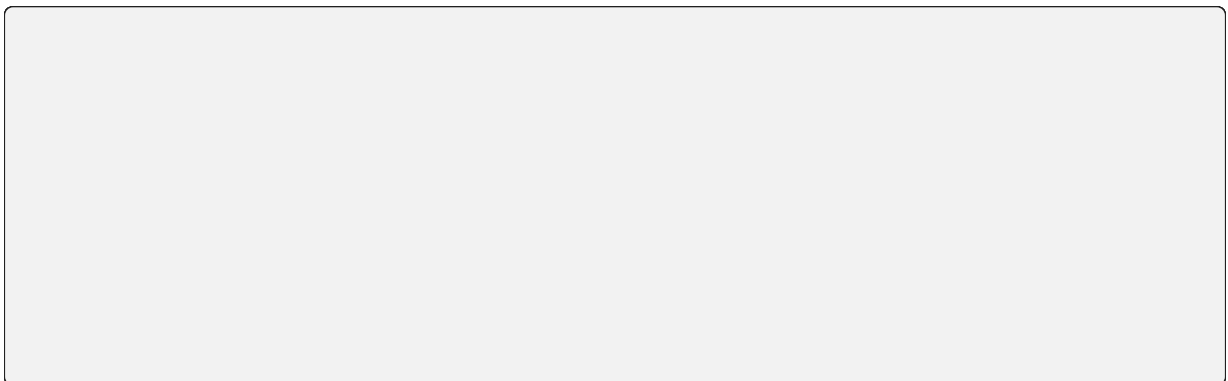
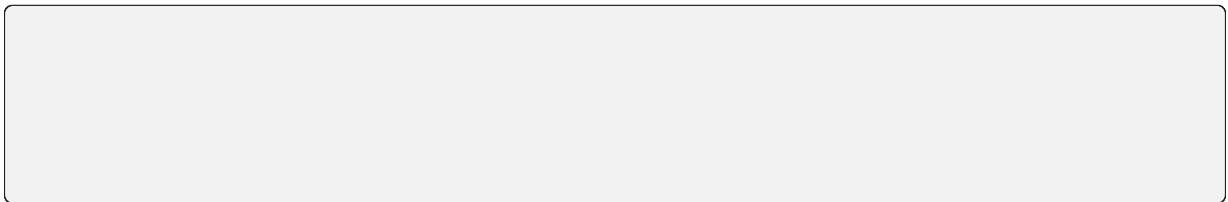
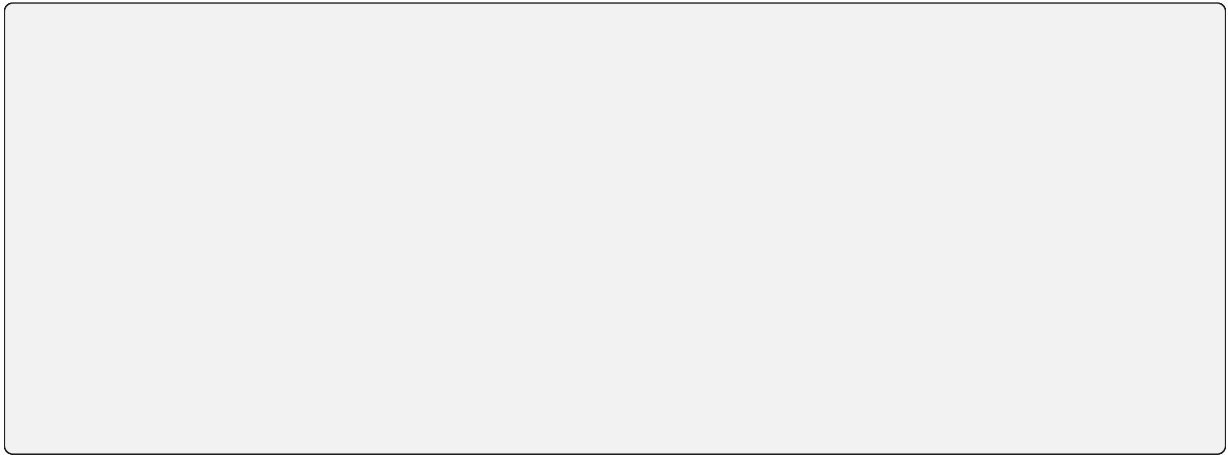
Parameter	Type	Description

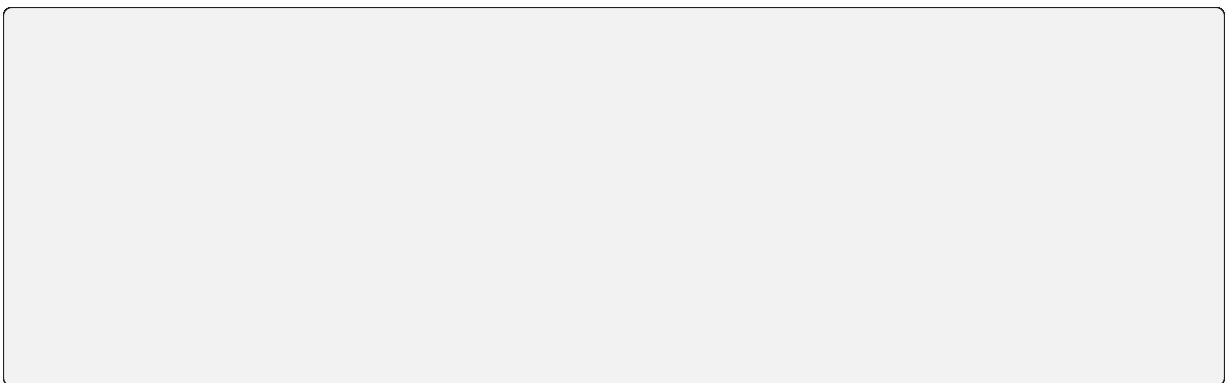
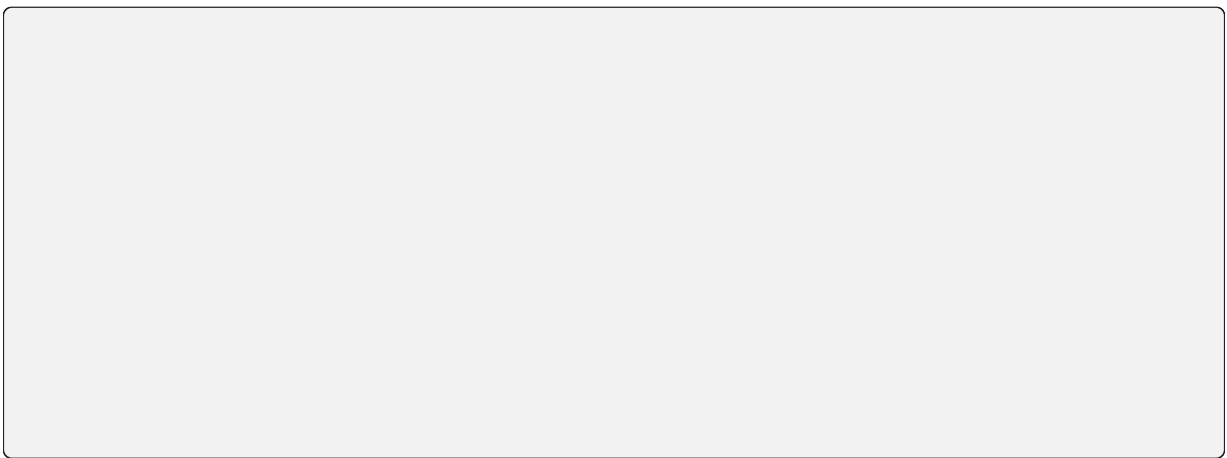
Response

HTTP Status Code	Description

Fields	Type	Values





Auto-completion

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

Fields	Type	Values

Interrupt

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

6.2.3 Code Execution (Batch Mode)

Uploading files

Parameters


Response

HTTP Status Code	Description



Executing with Build Step

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

Fields	Type	Values

Listing Files

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

Fields	Type	Values

Downloading Files

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

6.2.4 Code Execution (Streaming)

Code Execution

Terminal Emulation

Parameters

Parameter	Type	Description

Client-to-Server Protocol

Standard input stream

Terminal resize

**Ping**

**Restart**

**Server-to-Client Protocol**

**Standard output/error stream**

**Server-side errors**

## **6.2.5 Event Monitoring**

**Session Lifecycle Events**

Parameters

Parameter	Type	Description
		,
		,

Responses

Event Name	Description

Field Name	Description

Background Task Progress Events

Parameters

Parameter	Type	Description

Responses

Event Name	Description

Field Name	Type	Description

“”

6.2.6 Service Ports (aka Service Proxies)

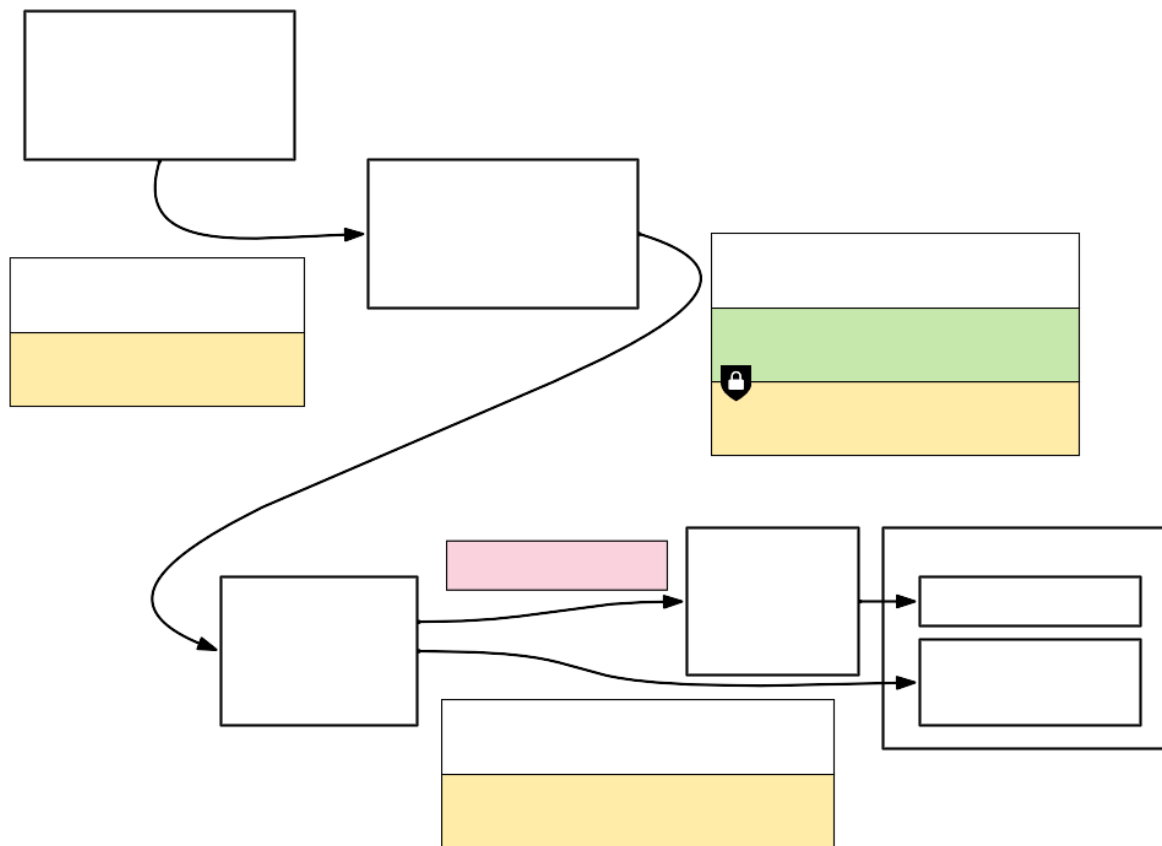
“”

,

,

---

---



Service Proxy (HTTP)

Parameters

Parameter	Type	Description

Service Proxy (TCP)

Parameters

Parameter	Type	Description

6.2.7 Resource Presets

Listing Resource Presets



Parameters

Response

HTTP Status Code	Description

Fields	Type	Values

Checking Allocatability of Resource Presets

,

Parameters

Response

HTTP Status Code	Description

Fields	Type	Values
		“”.
		“”.
		“”
		”

6.2.8 Virtual Folders

Listing Virtual Folders

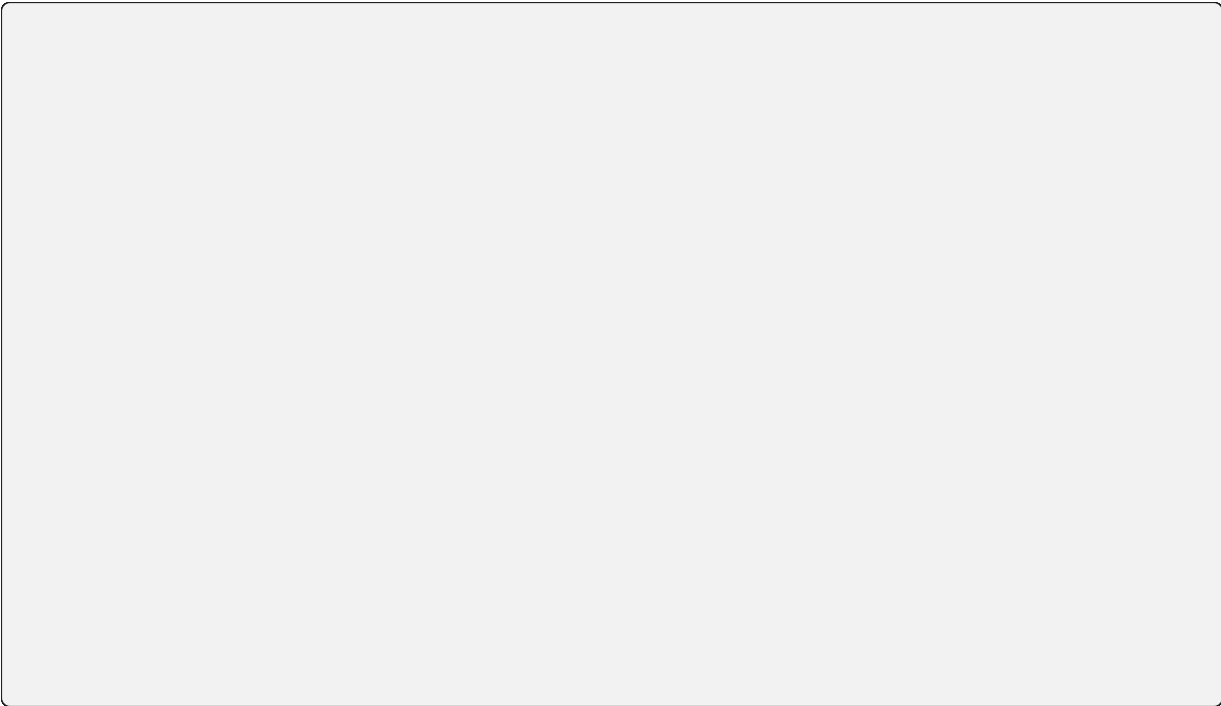
Parameters

Parameter	Typ	Description

Response

HTTP Status Code	Description

Fields	Typ	Values



Listing Virtual Folder Hosts

Parameters

Response

HTTP Status Code	Description

Fields	Type	Values

Creating a Virtual Folder

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

Fields	Type	Values

Getting Virtual Folder Information

## Parameters

Parameter	Type	Description

## Response

HTTP Status Code	Description

Fields	Type	Values

## Deleting Virtual Folder

## Parameters

Parameter	Description

## Response

HTTP Status Code	Description

## Rename a Virtual Folder

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

Listing Files in Virtual Folder

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

Fields	Type	Values

Uploading a File to Virtual Folder

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

Fields	Type	Values

Creating New Directory in Virtual Folder

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

Downloading a File or a Directory from a Virtual Folder

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

Fields	Type	Values

Deleting Files in Virtual Folder

Parameters

Parameter	Type	Description



## Response

HTTP Status Code	Description

## Rename a File in Virtual Folder

## Parameters

Parameter	Type	Description

## Response

HTTP Status Code	Description

## Listing Invitations for Virtual Folder

## Parameters

## Response

HTTP Status Code	Description

Fields	Type	Values

Creating an Invitation

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

Fields	Type	Values

Accepting an Invitation

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

## Rejecting an Invitation

### Parameters

Parameter	Type	Description

### Response

HTTP Status Code	Description

Fields	Type	Values

## Listing Sent Invitations

### Parameters

### Response

HTTP Status Code	Description

Fields	Type	Values

Updating an Invitation

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

Fields	Type	Values

Leave an Shared Virtual Folder

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

Fields	Type	Values

Listing Users Share Virtual Folders

,

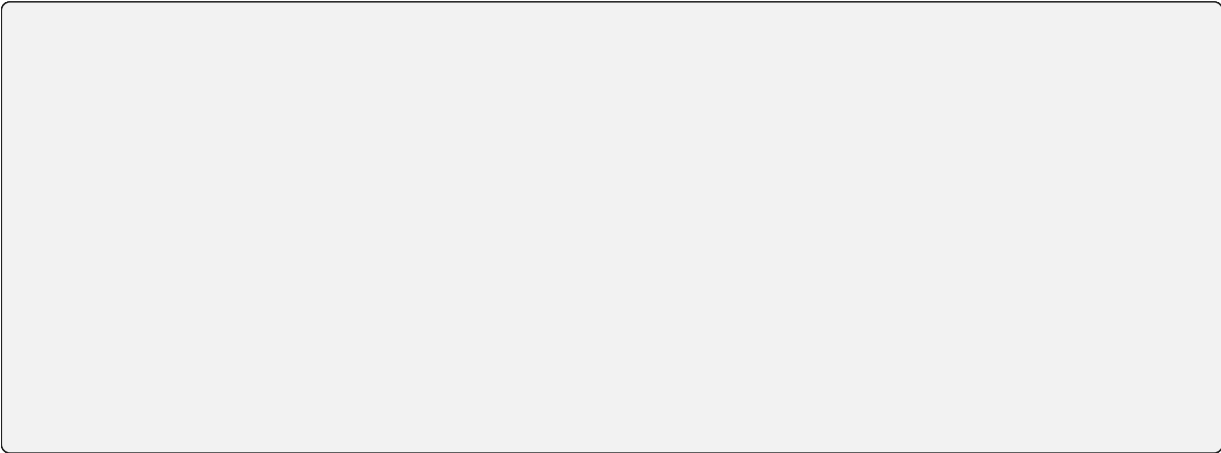
Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

Fields	Type	Values



Updating the permission of a shared virtual folder

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

Fields	Type	Values

Share a Group Virtual Folder to an Individual Users

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

Fields	Type	Values

Unshare a Group Virtual Folder from Users

Parameters

Parameter	Type	Description

Response

HTTP Status Code	Description

Fields	Type	Values

Clone a Virtual Folder

Parameters

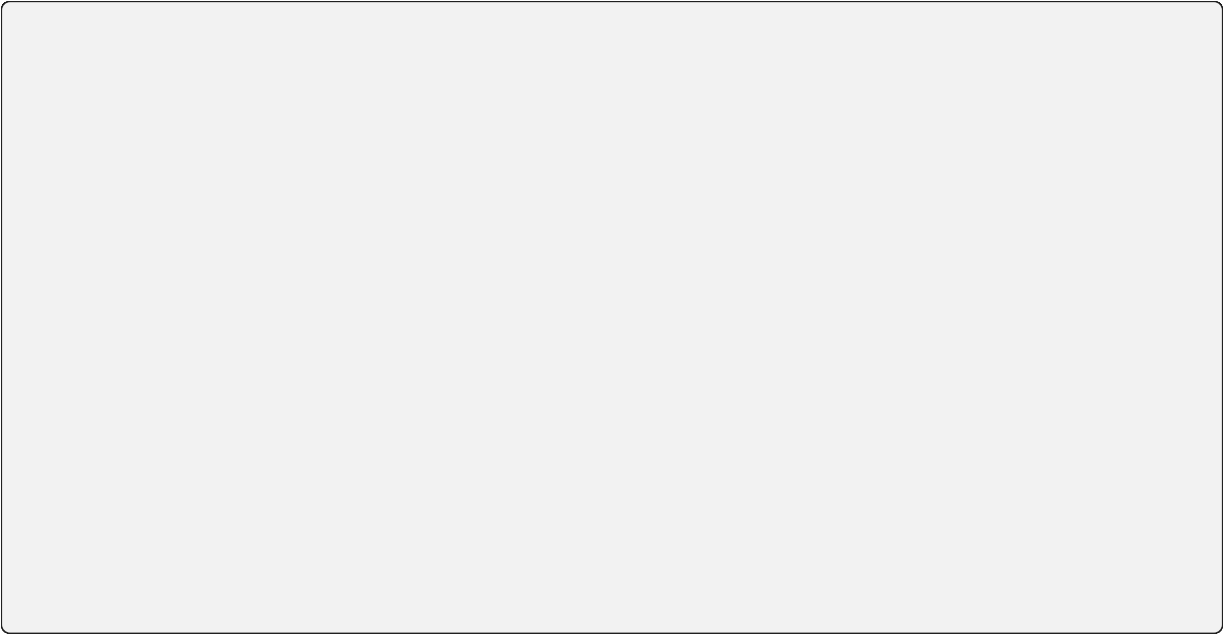
Parameter	Type	Description

Response

HTTP Status Code	Description

Fields	Type	Values

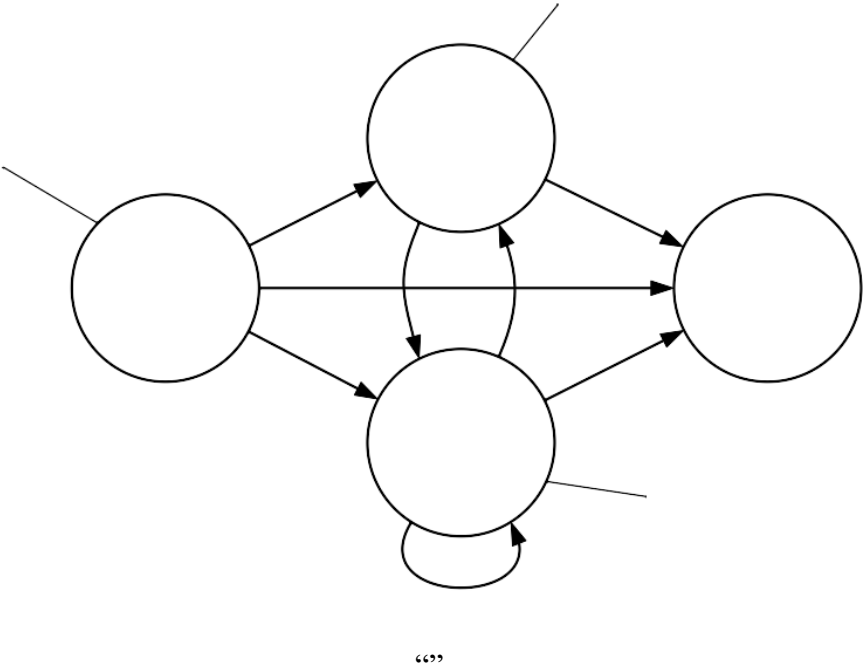
Fields	Type	Values



6.2.9 Code Execution Model

,





,

### 6.2.10 Handling Console Output

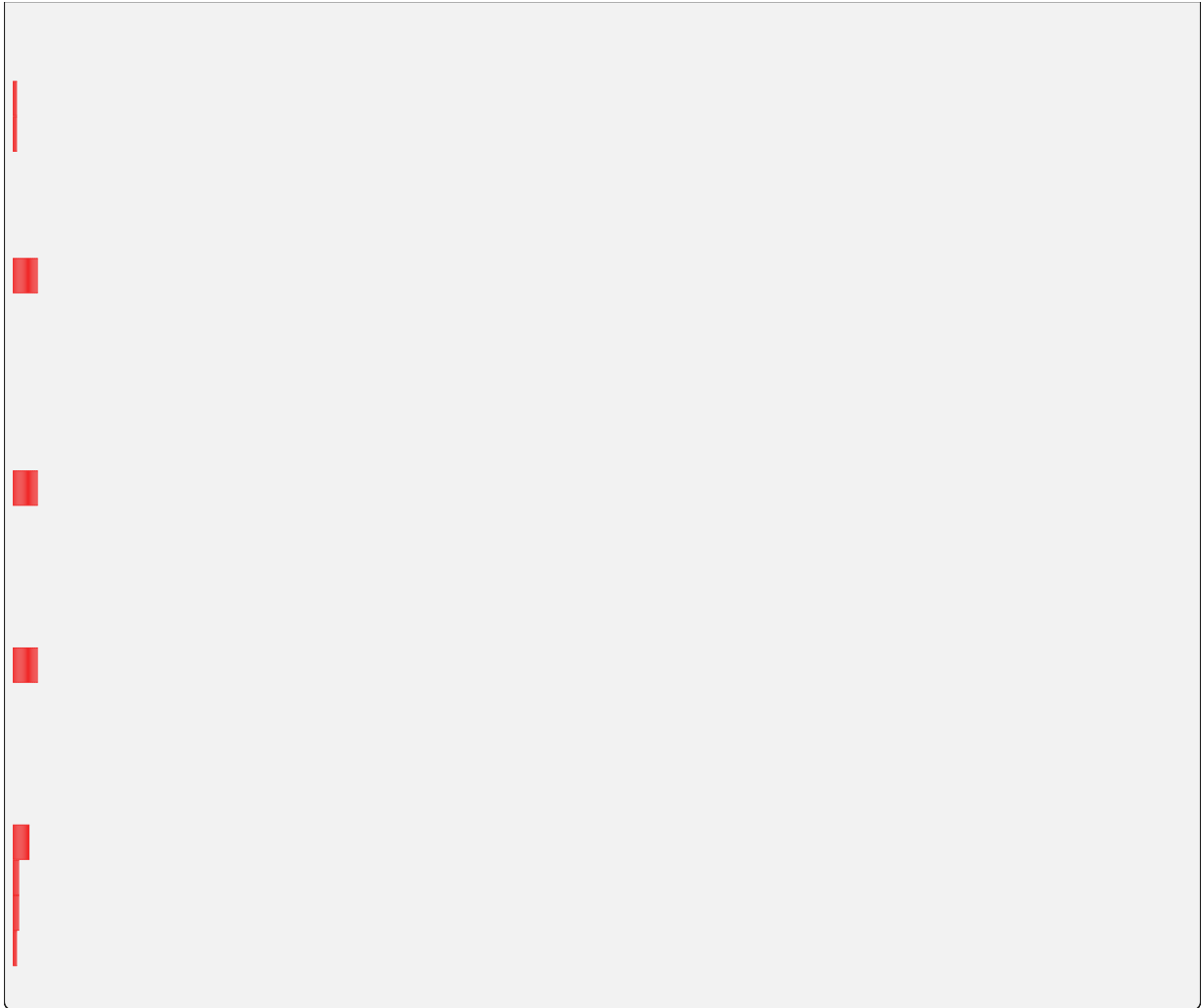
,

## 6.3 Manager GraphQL API

### 6.3.1 Domain Management

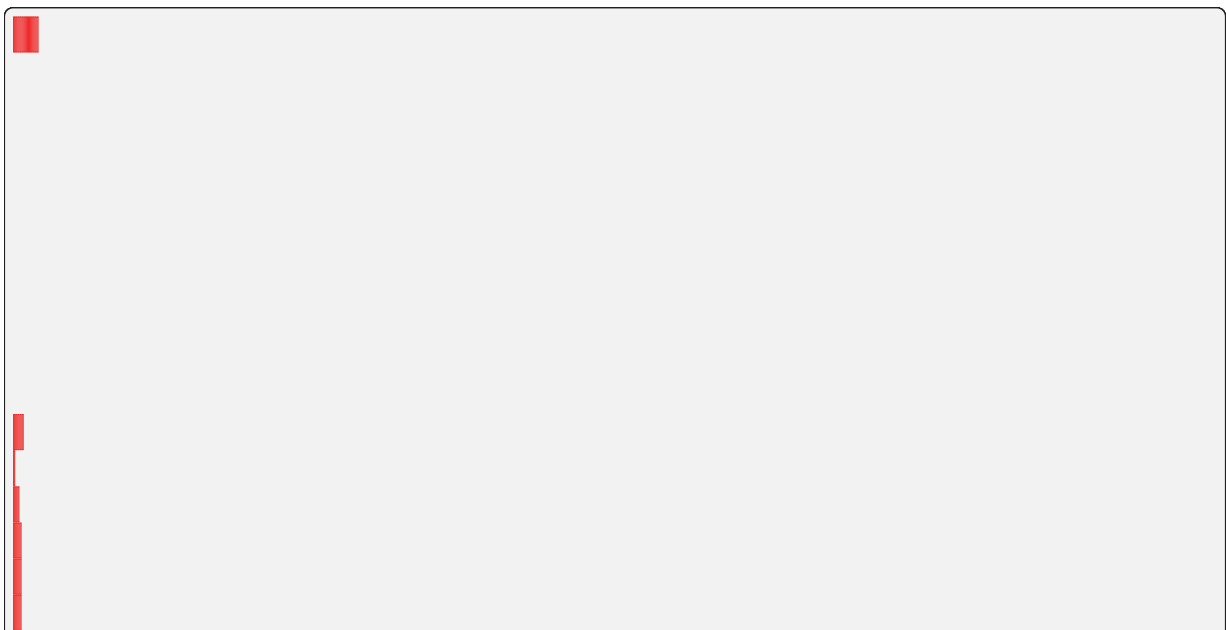
#### Query Schema

#### Mutation Schema



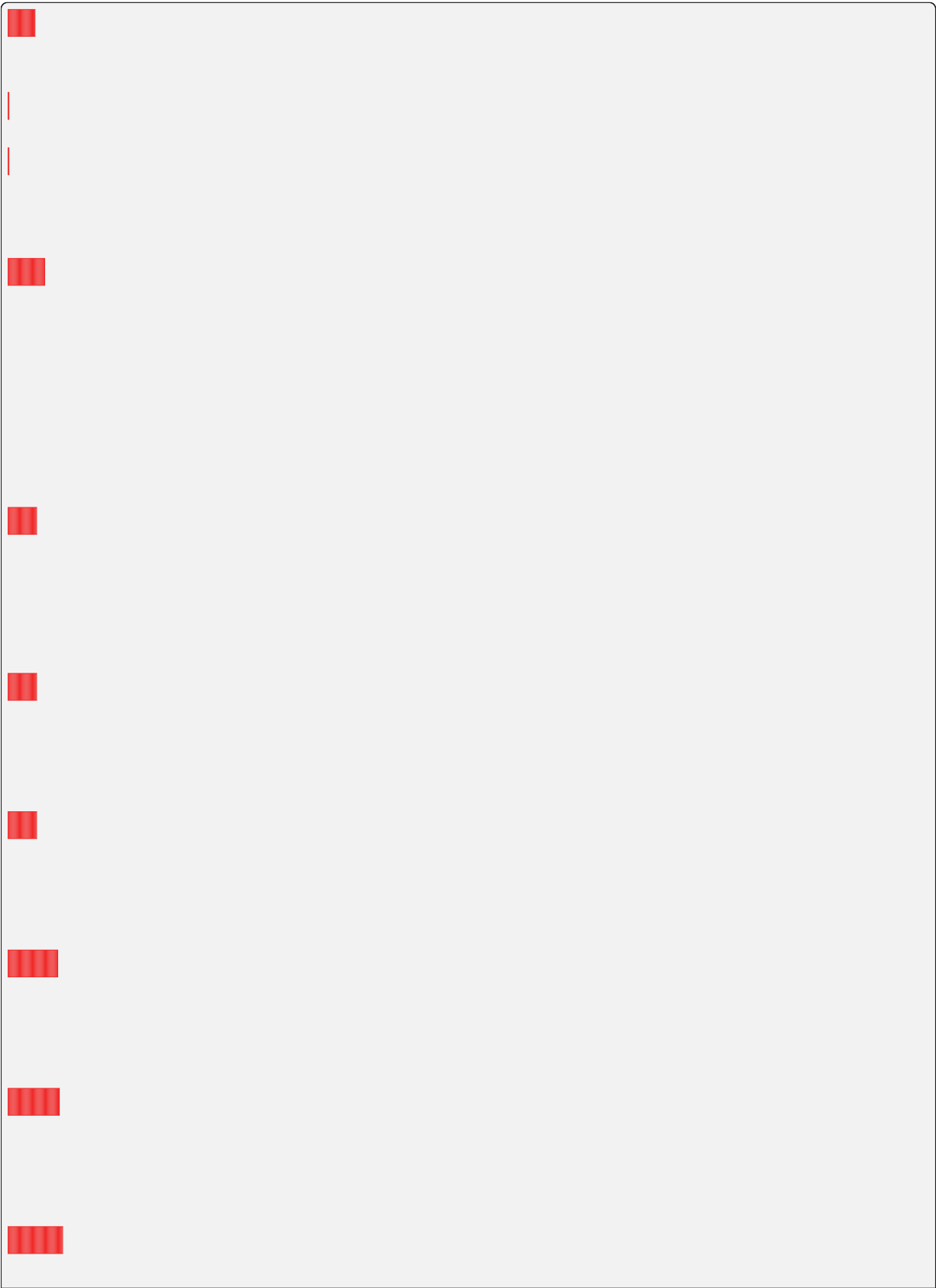
## 6.3.2 Scaling Group Management

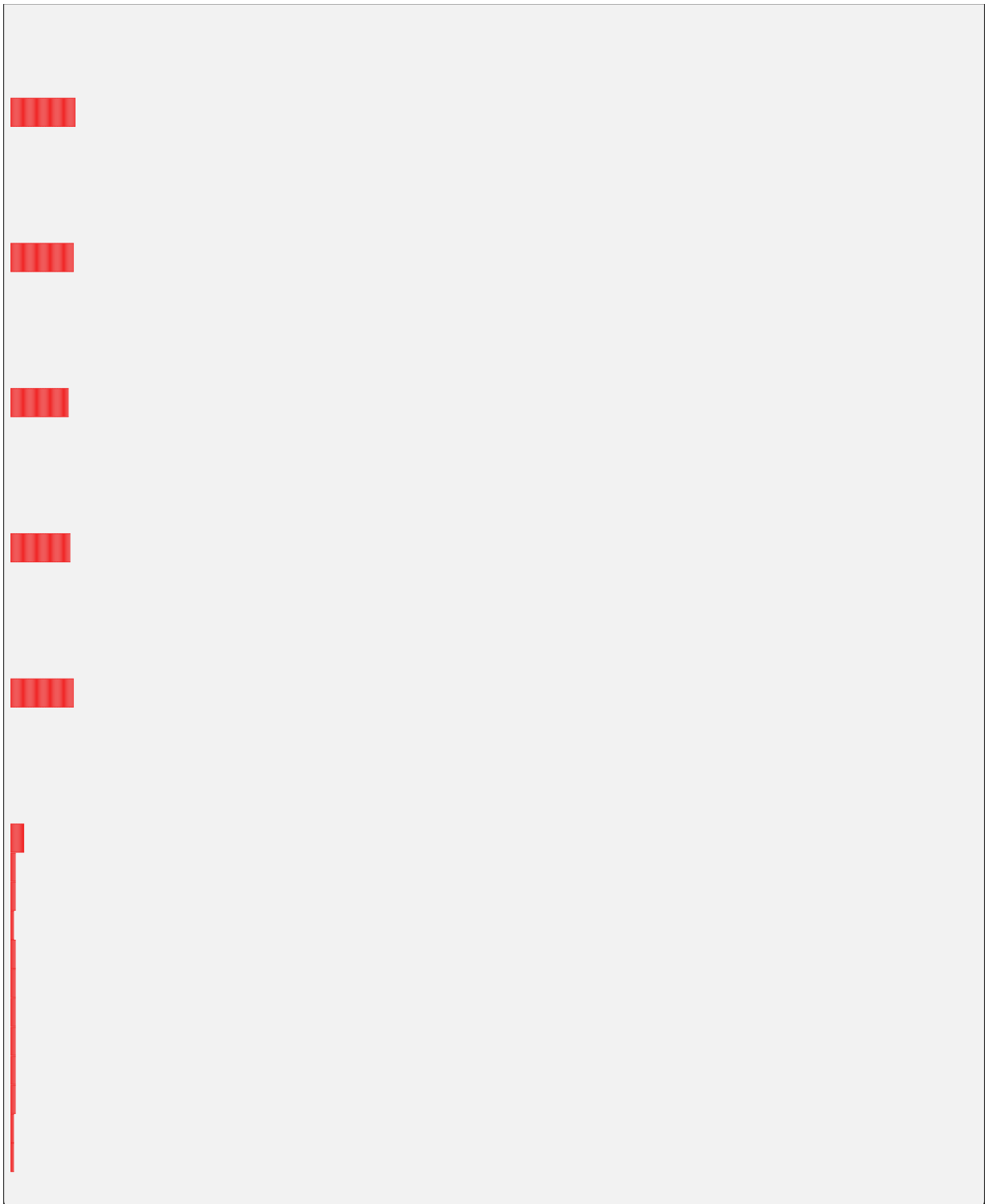
### Query Schema





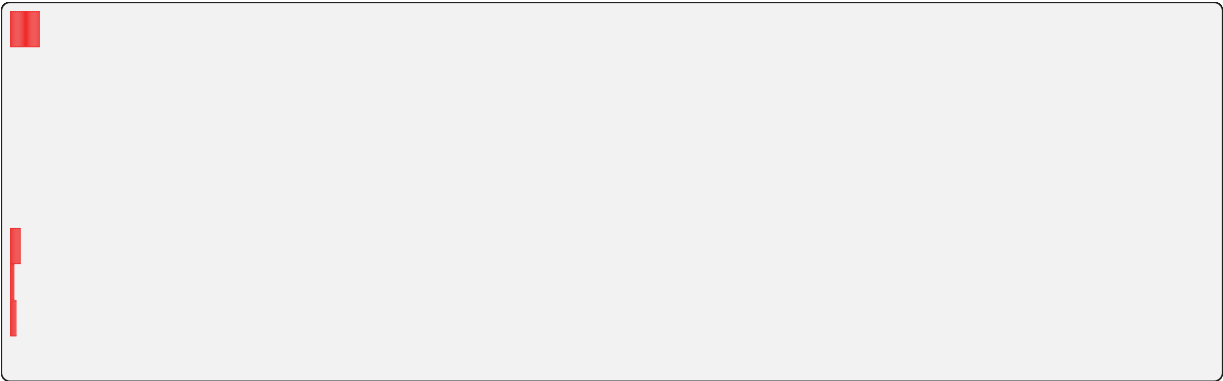
**Mutation Schema**





### 6.3.3 Resource Preset Management

#### Query Schema

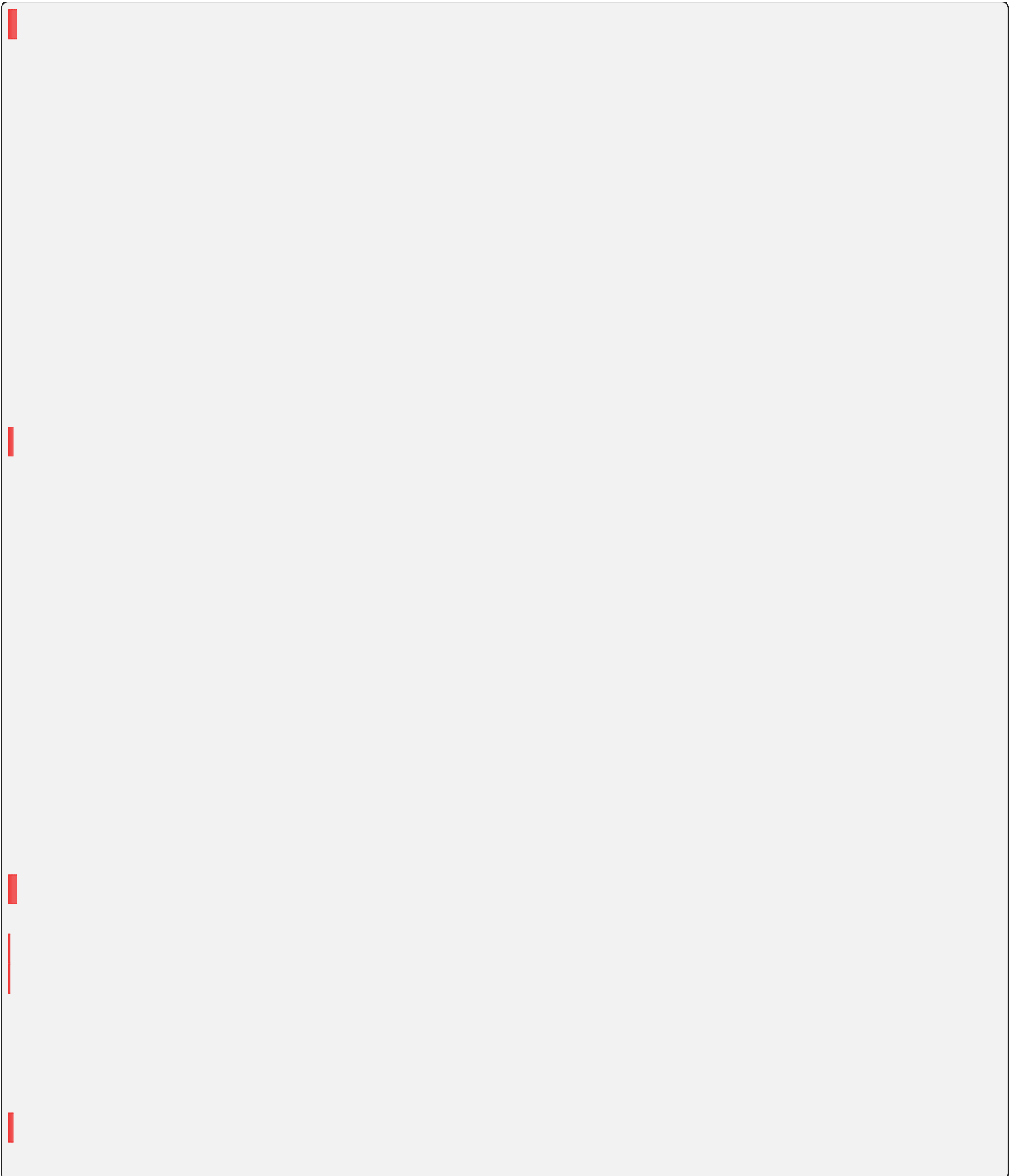


#### Mutation Schema



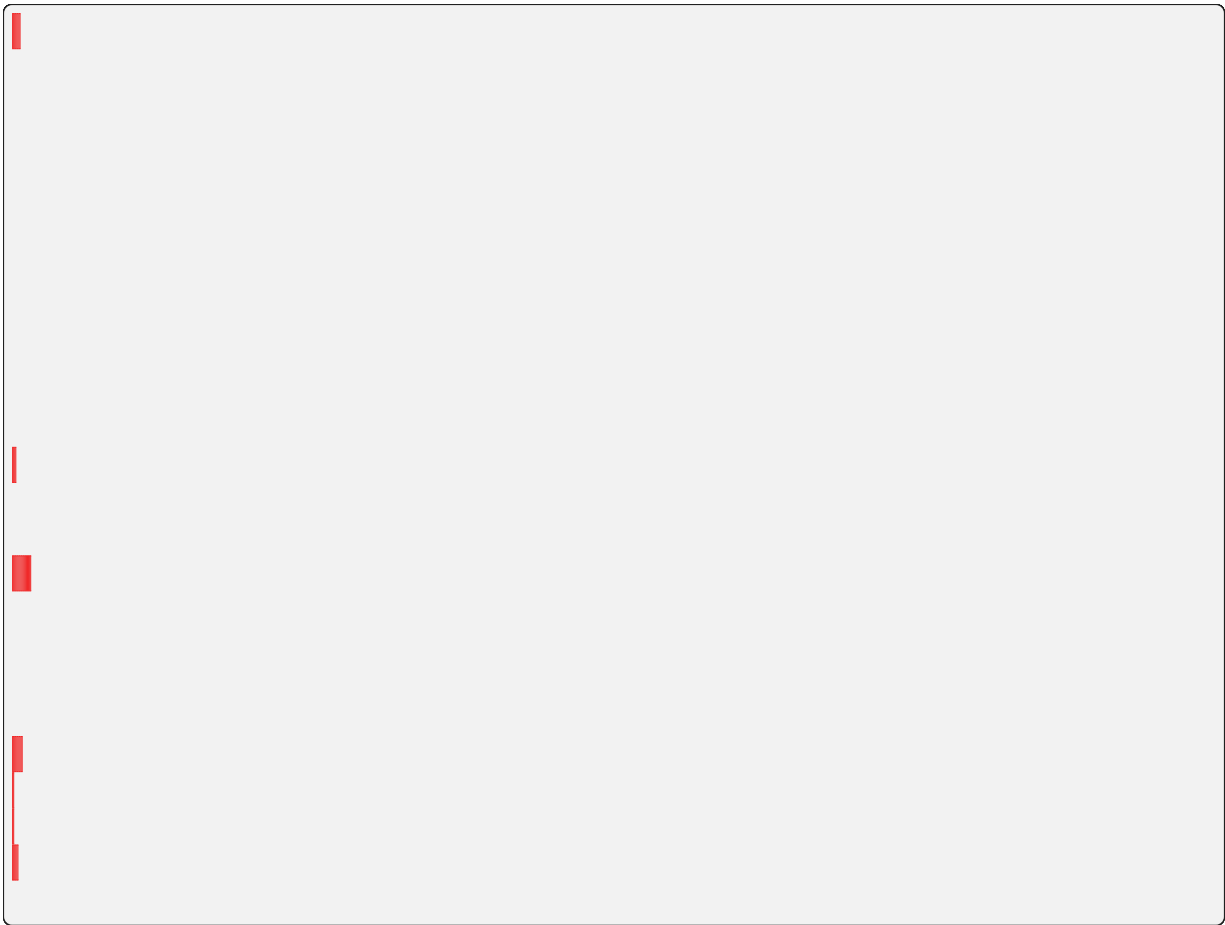
## 6.3.4 Agent Monitoring

### Query Schema

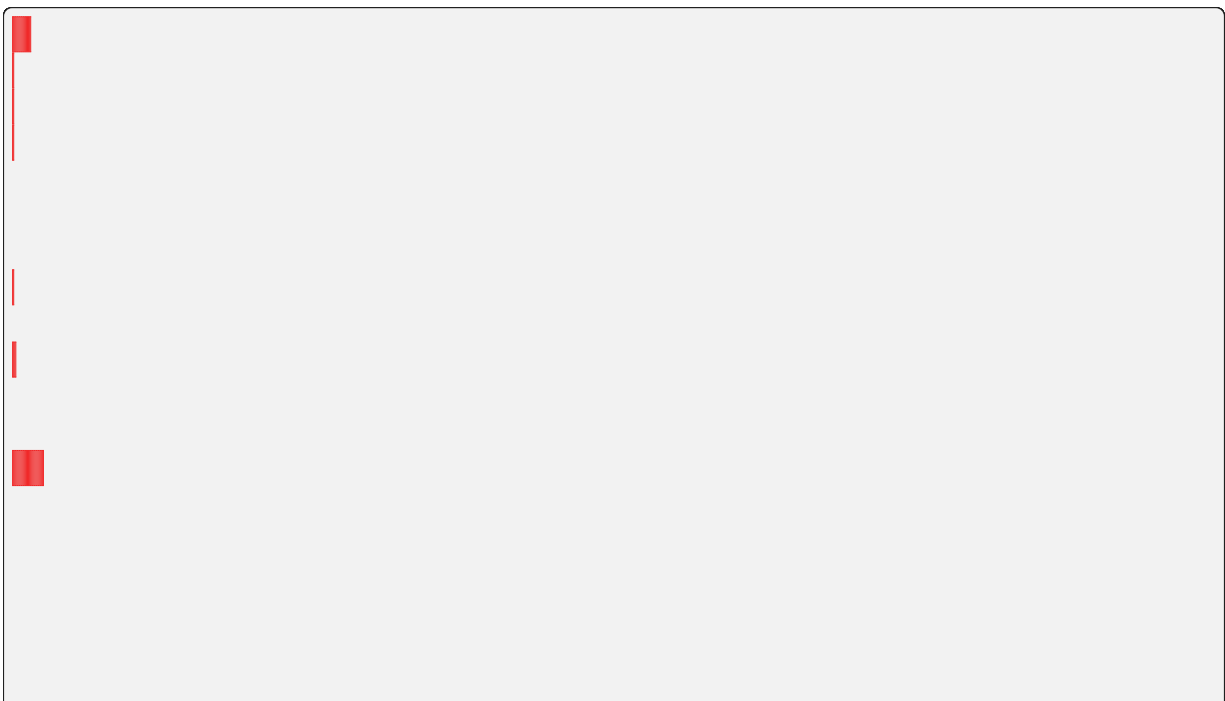


## 6.3.5 User Management

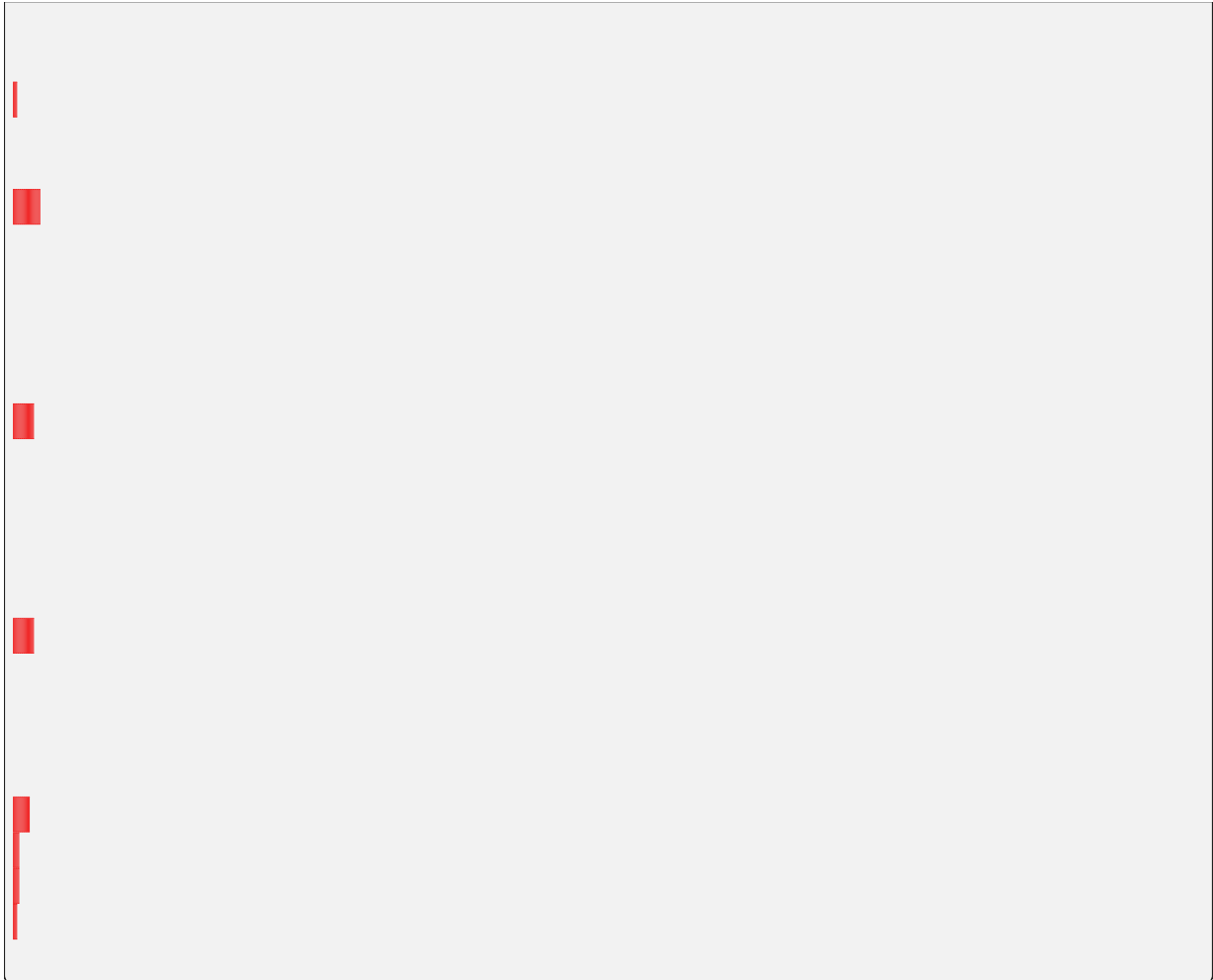
### Query Schema



### Mutation Schema

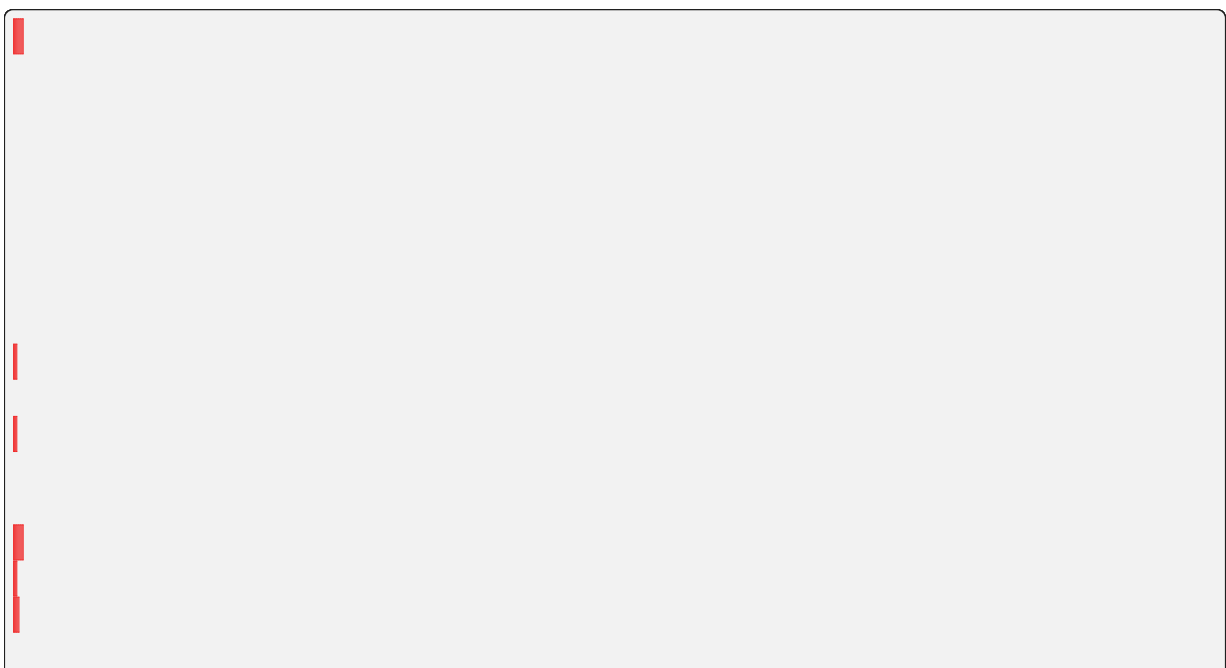






### 6.3.6 Group Management

#### Query Schema

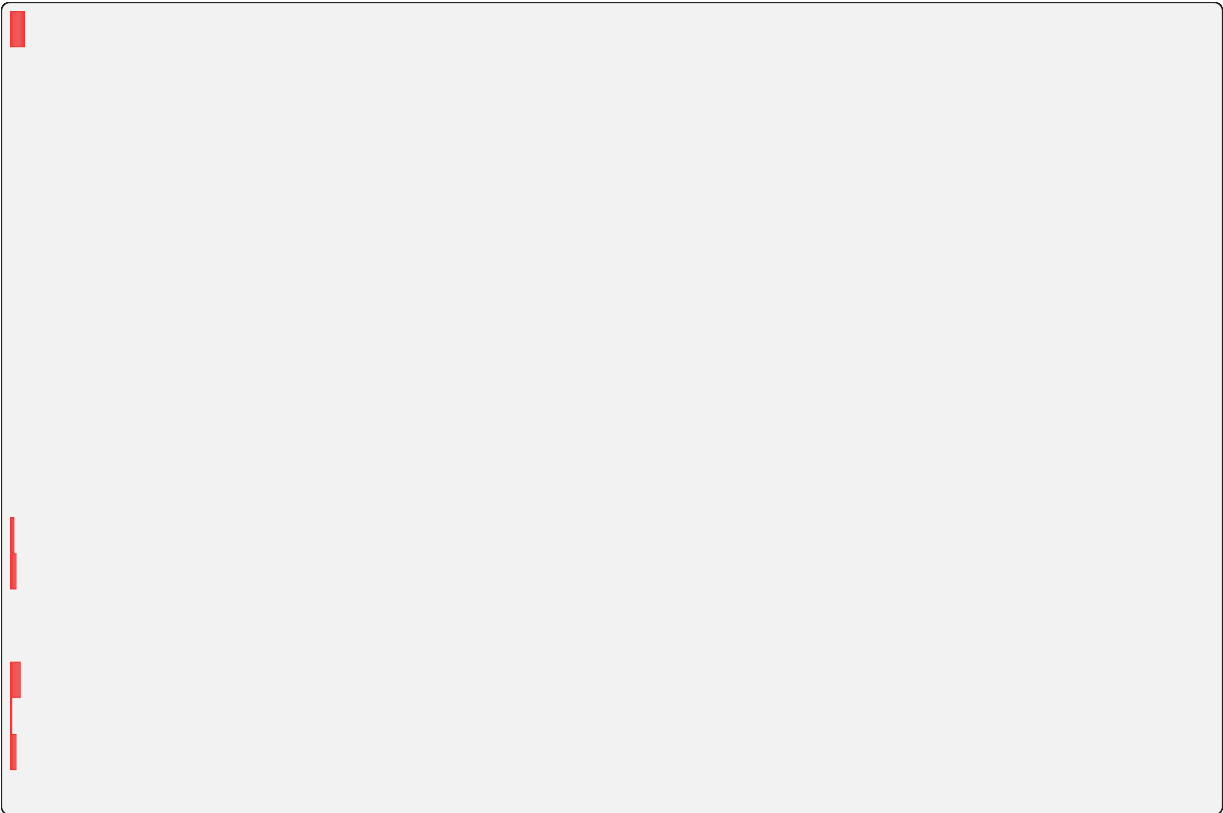


Mutation Schema

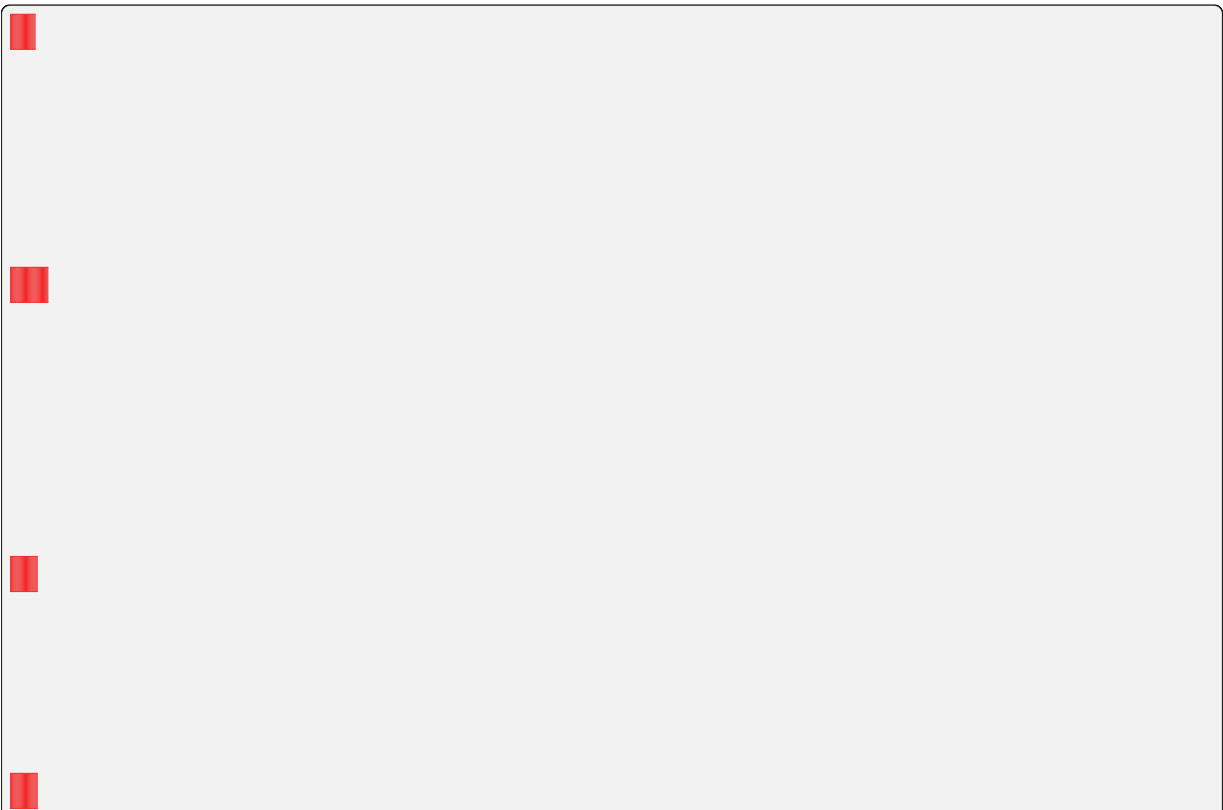


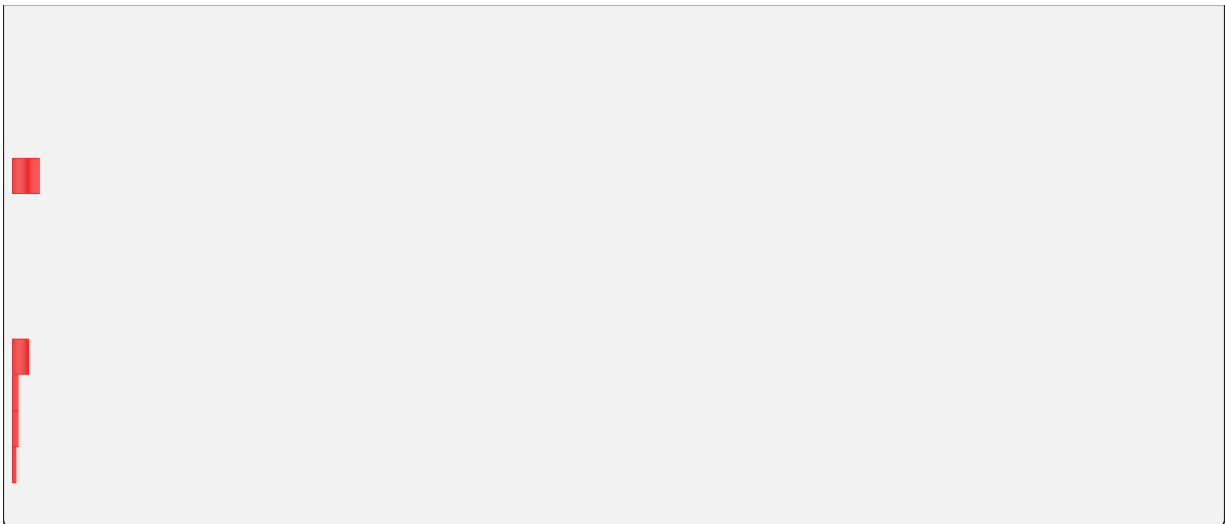
### 6.3.7 KeyPair Management

#### Query Schema



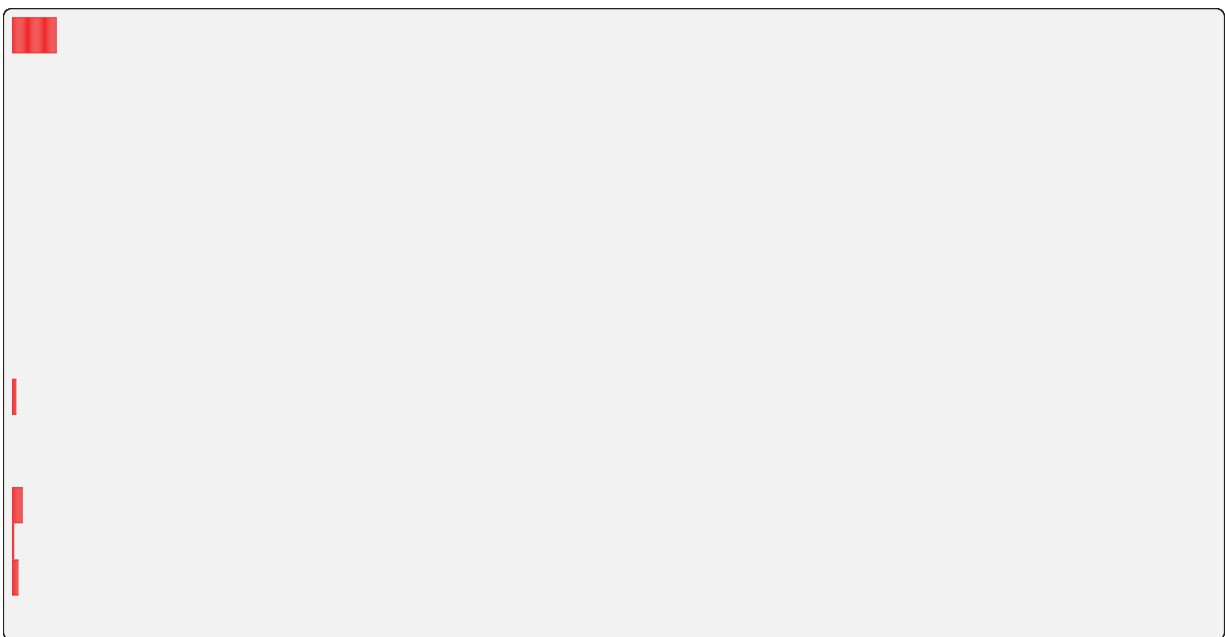
#### Mutation Schema



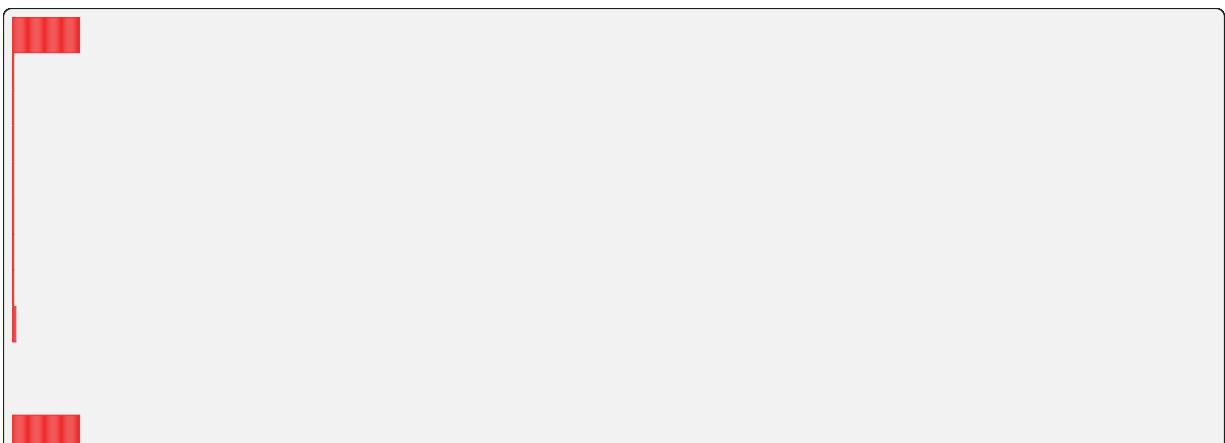


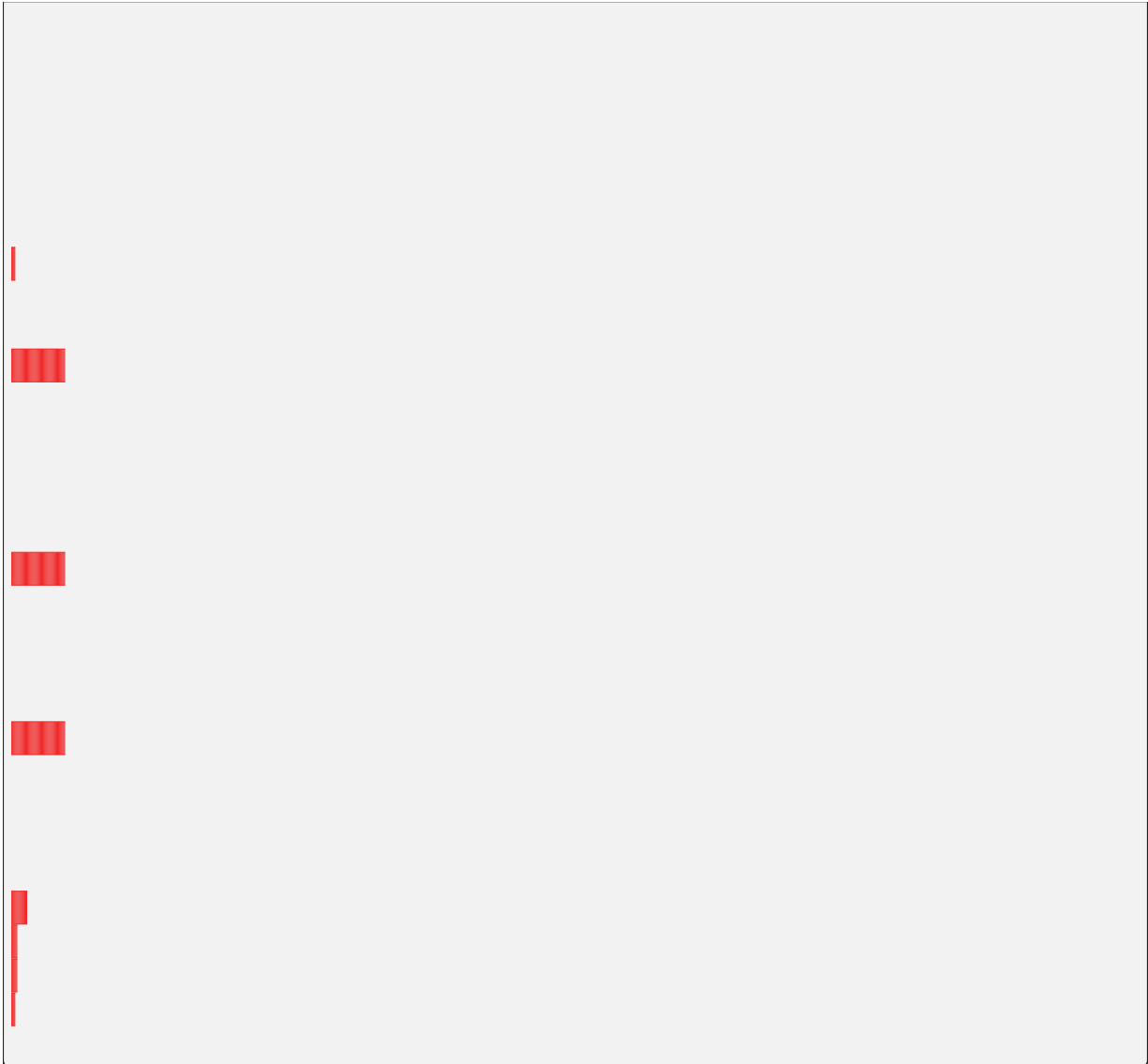
### 6.3.8 KeyPair Resource Policy Management

#### Query Schema



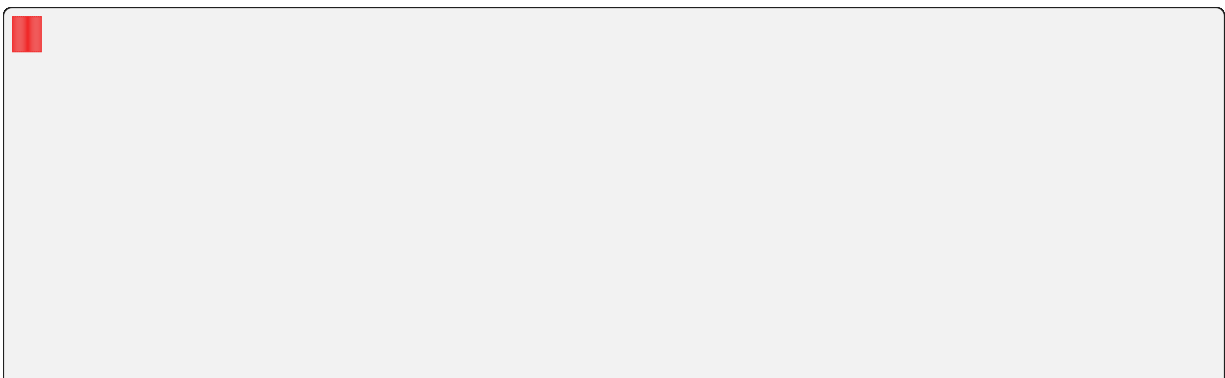
#### Mutation Schema

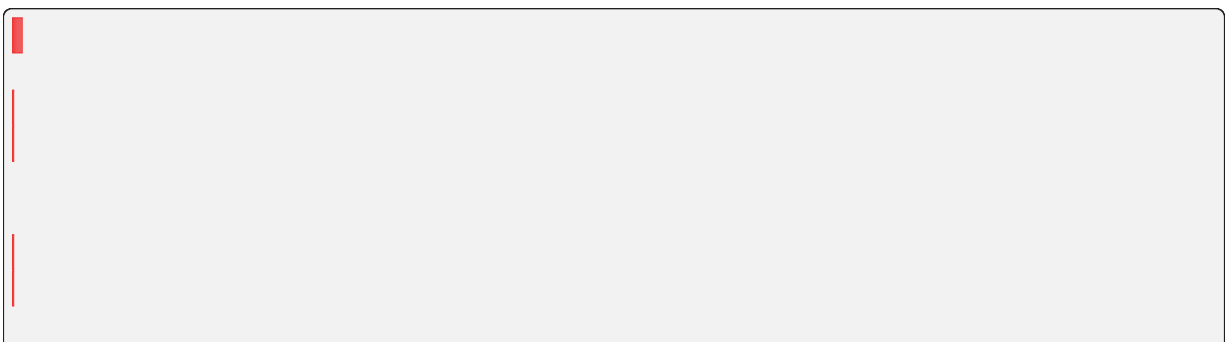
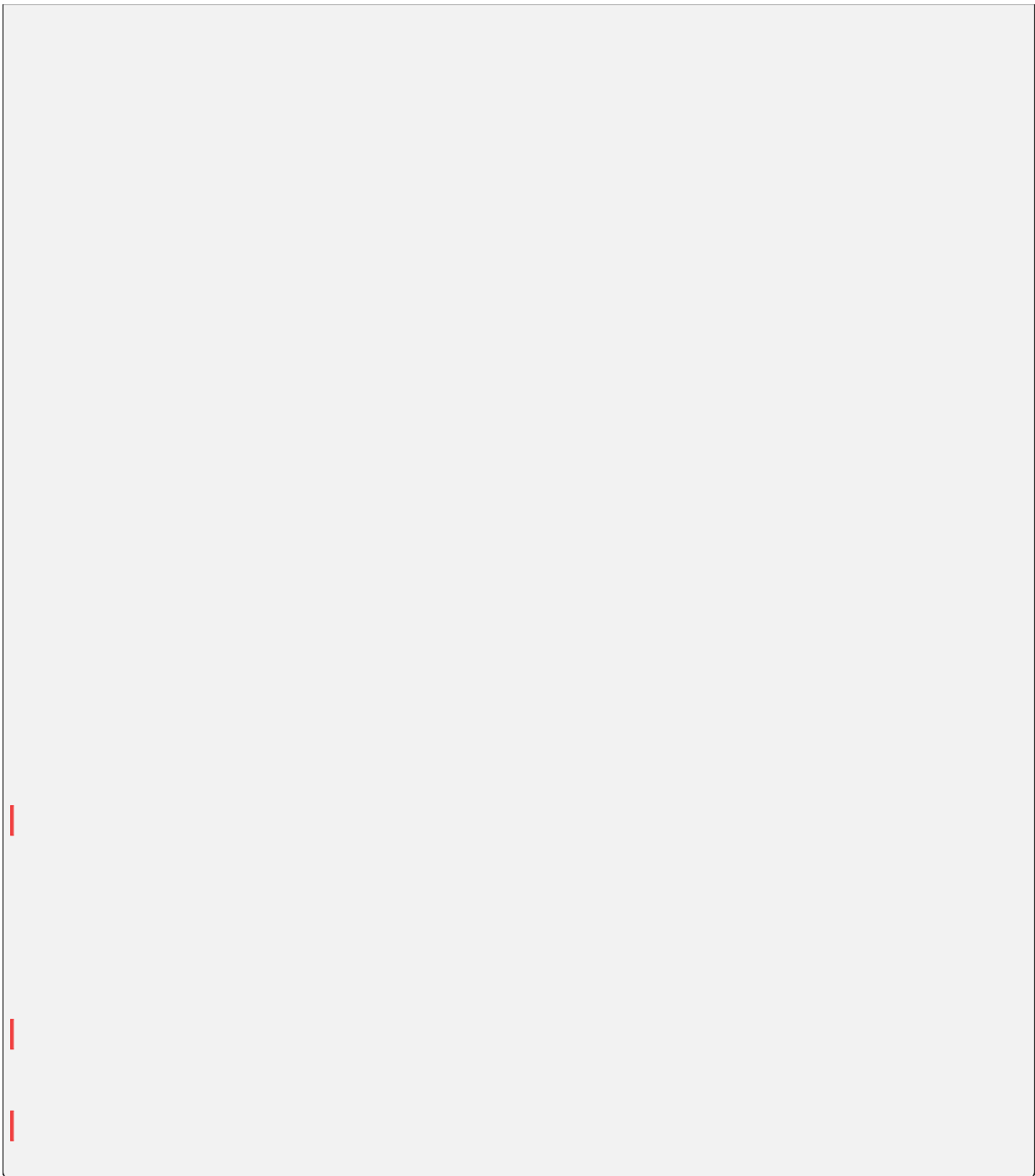


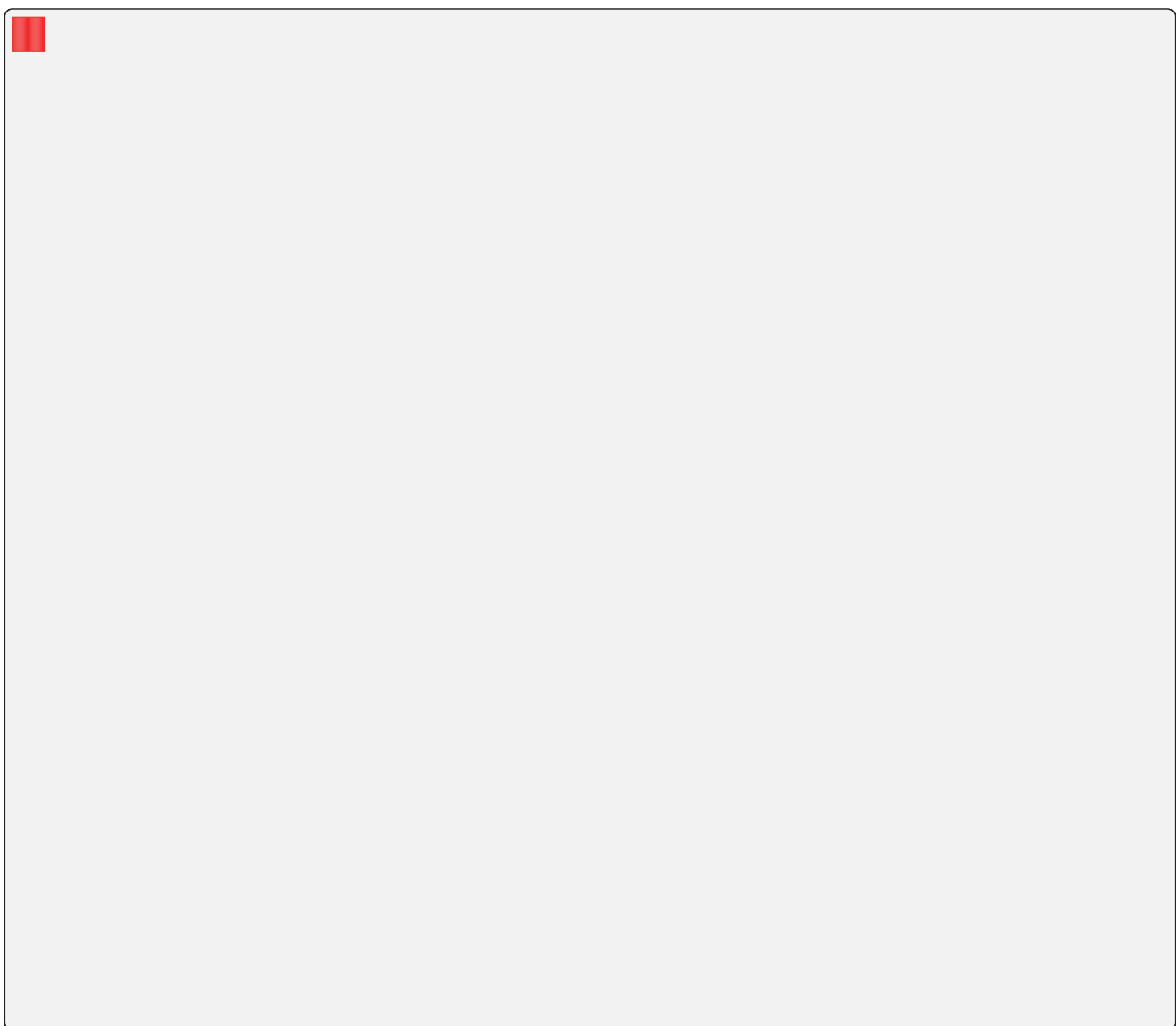
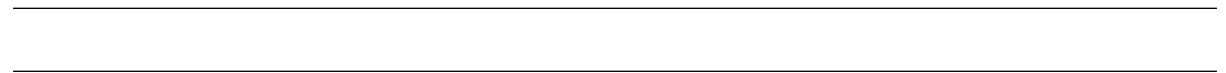
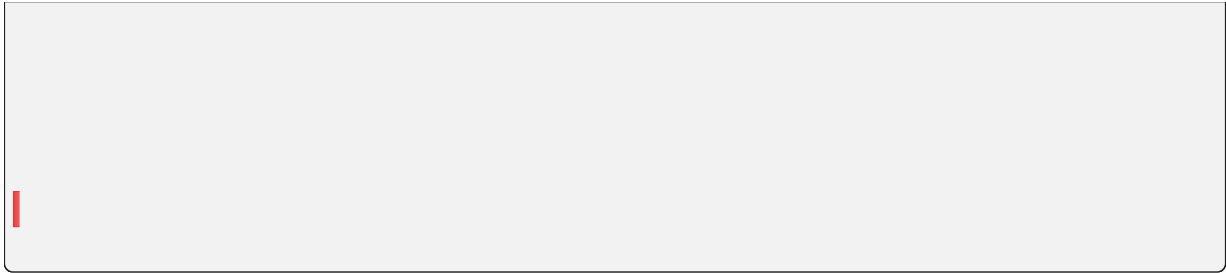


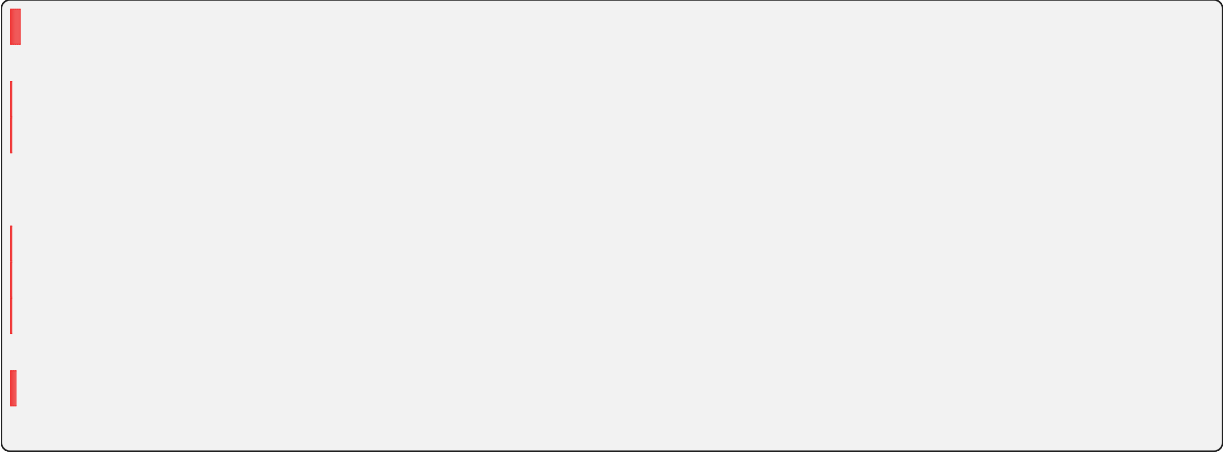
### 6.3.9 Compute Session Monitoring

#### Query Schema

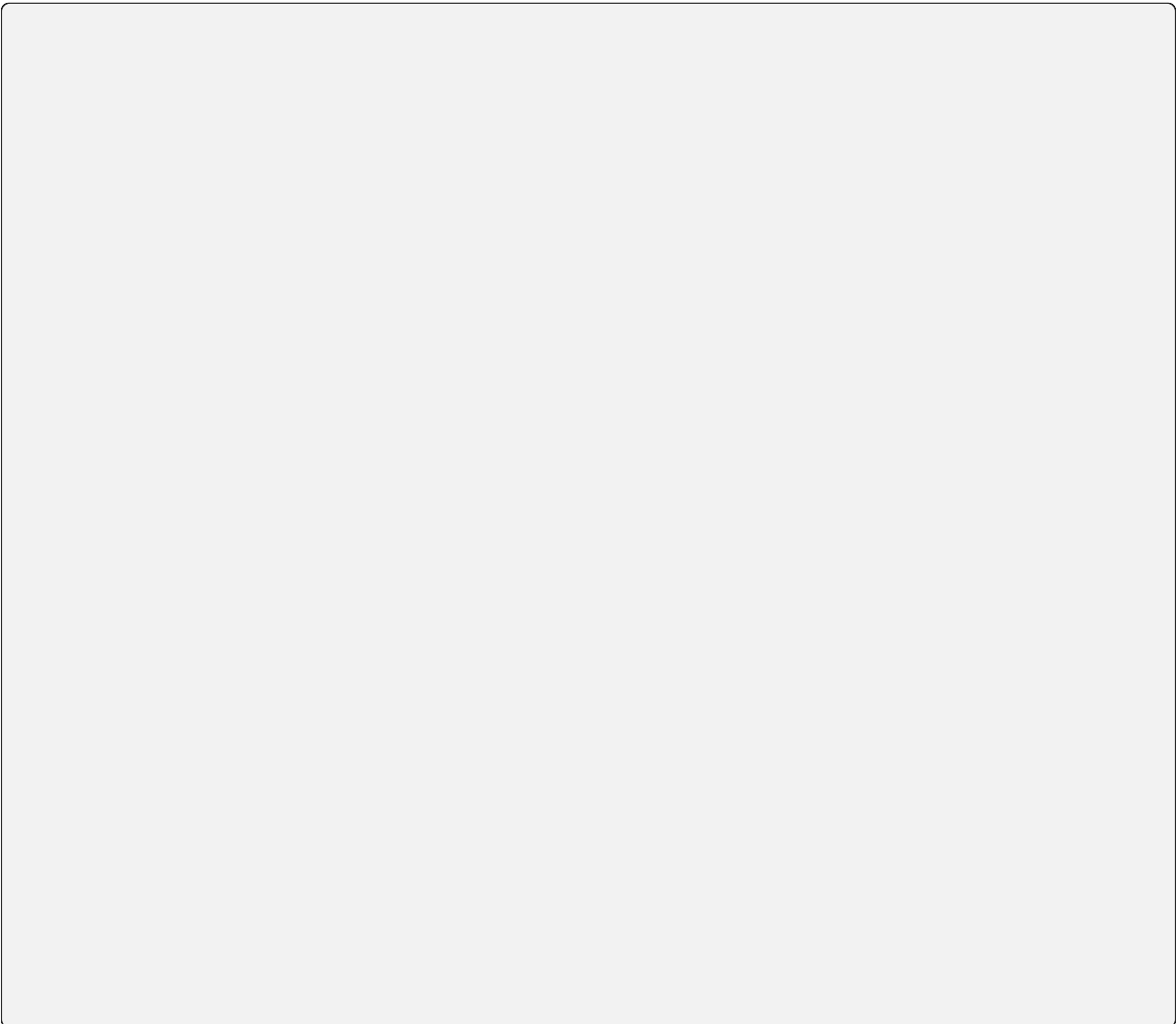






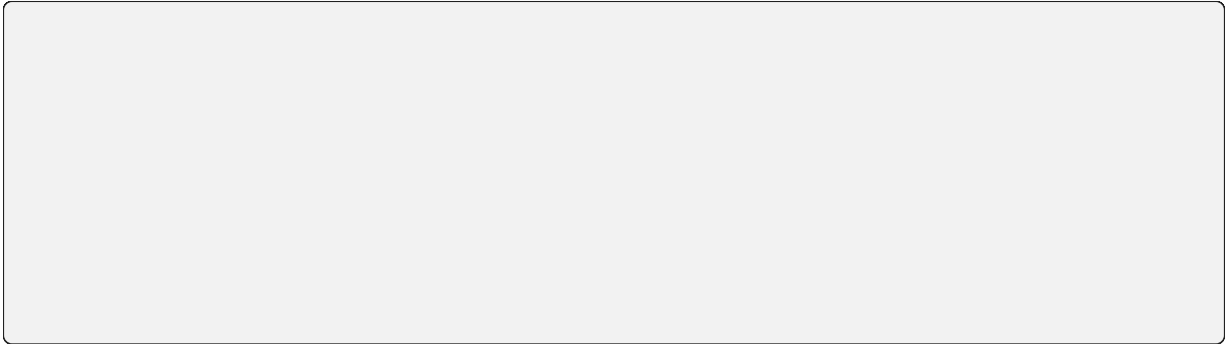


### Query Example

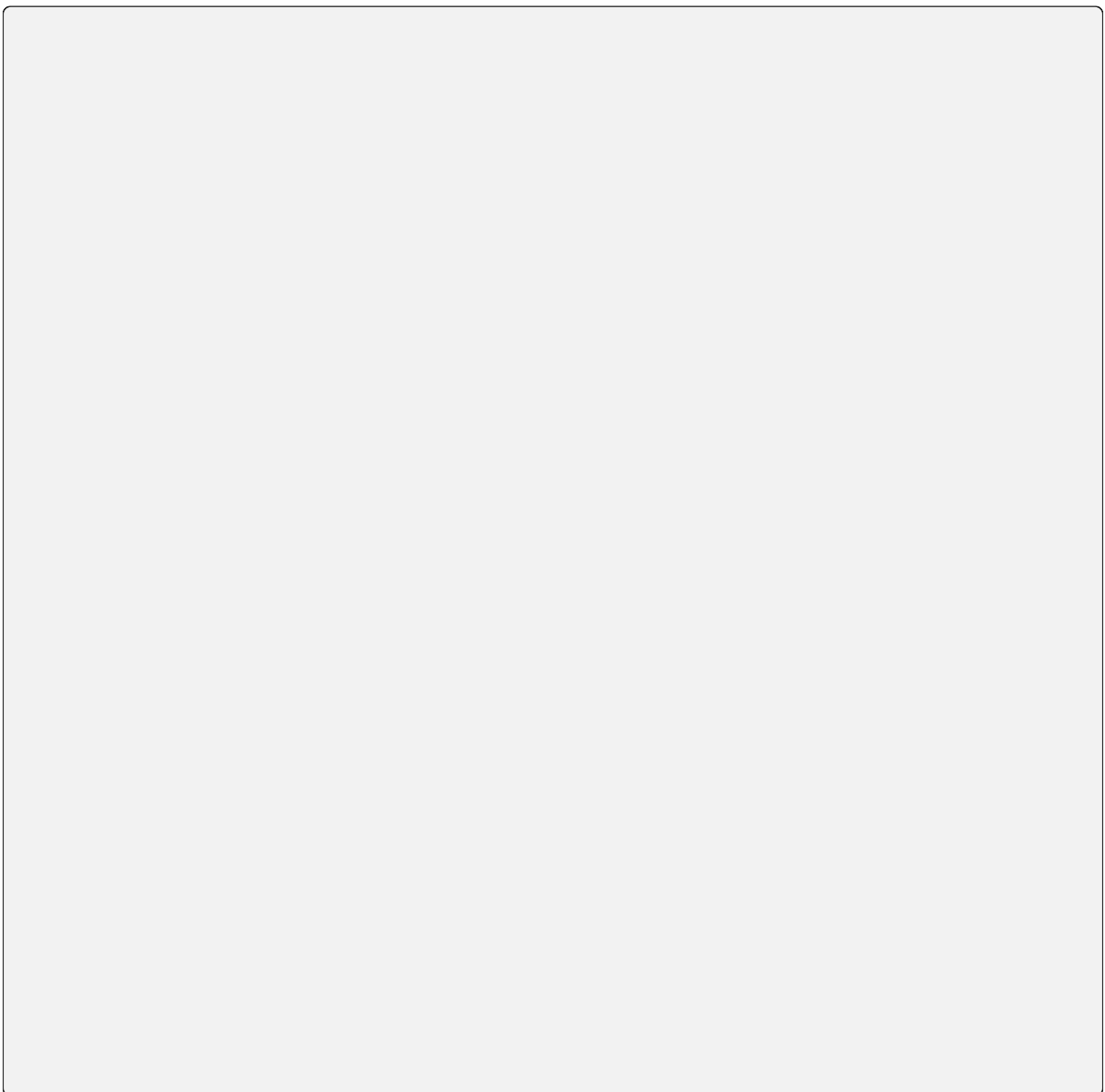




### API Parameters

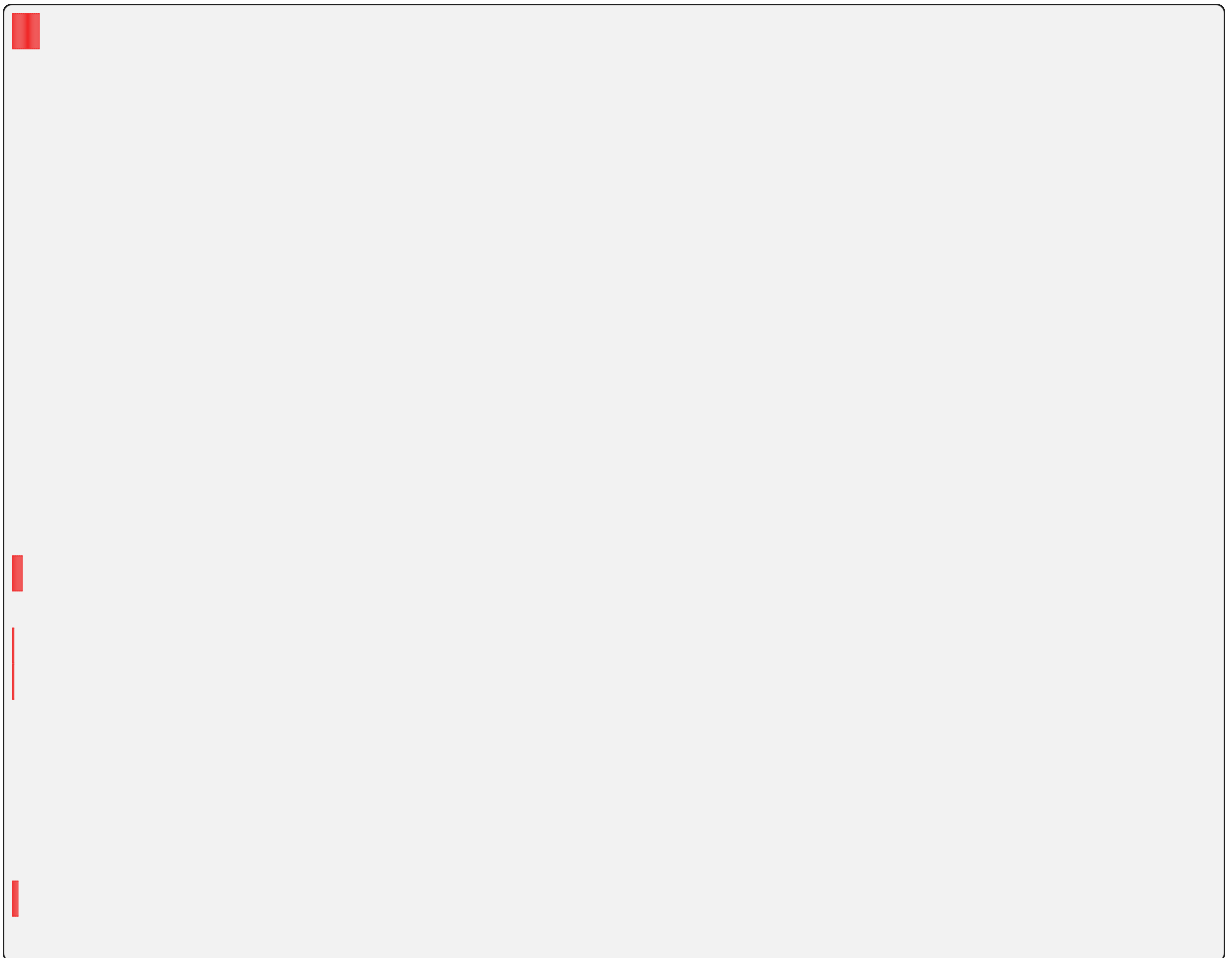
A large, empty rectangular box with a light gray background and a thin black border, intended for listing API parameters.

### API Response

A large, empty rectangular box with a light gray background and a thin black border, intended for showing the API response.

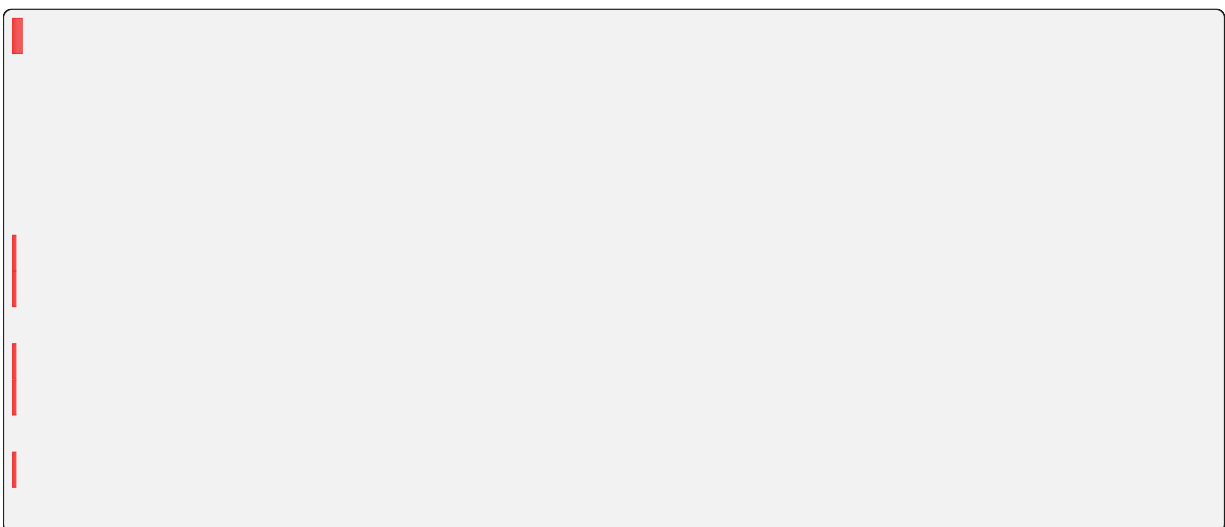
## 6.3.10 Virtual Folder Management

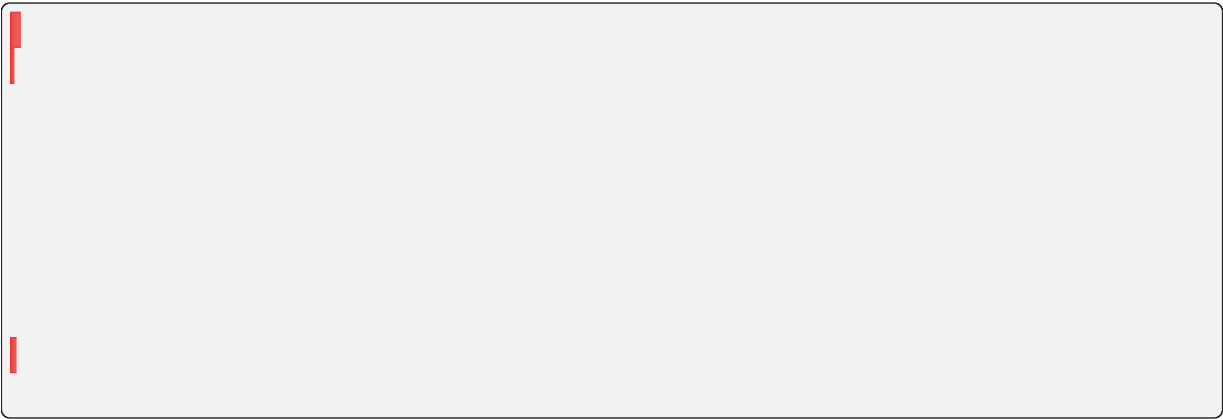
### Query Schema



## 6.3.11 Image Management

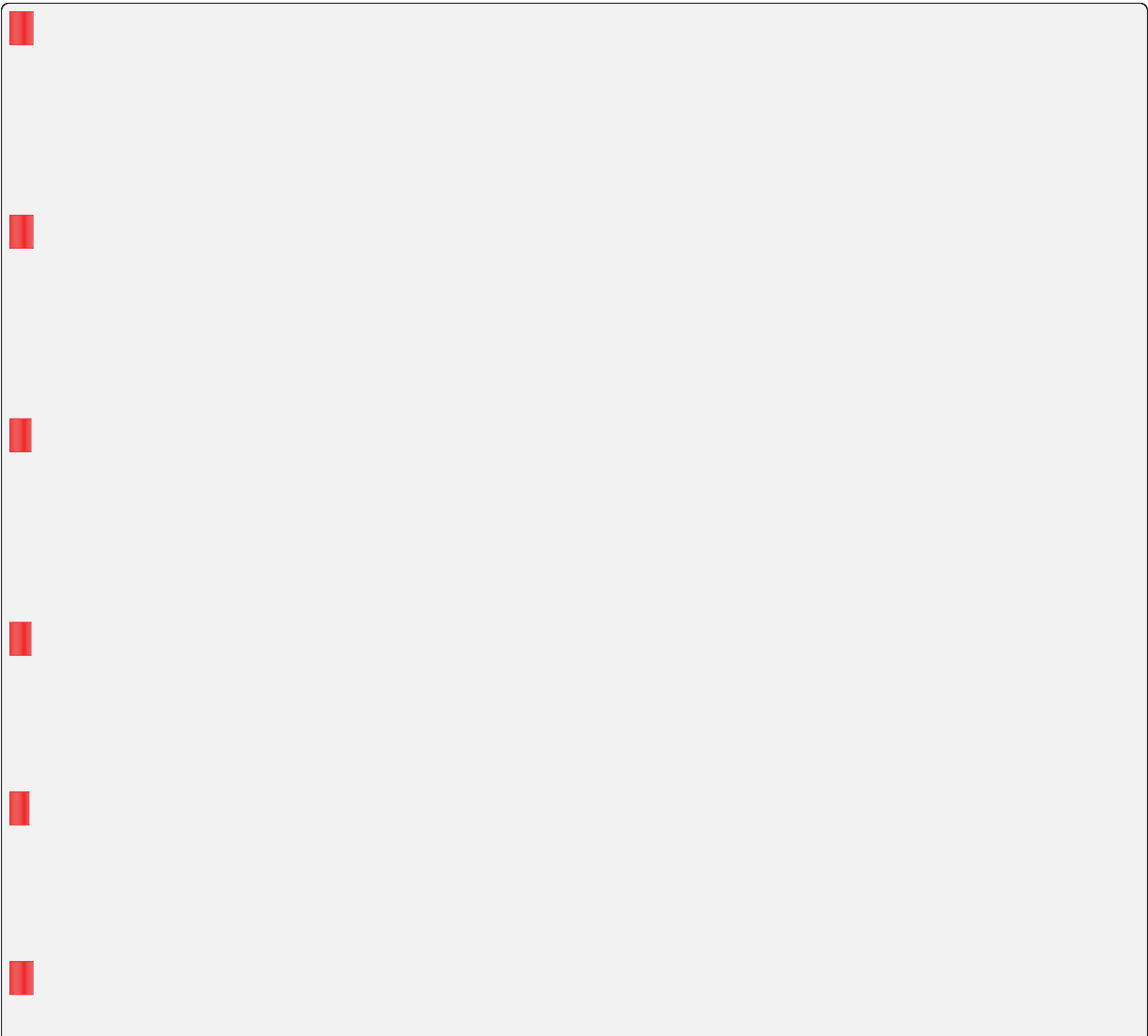
### Query Schema

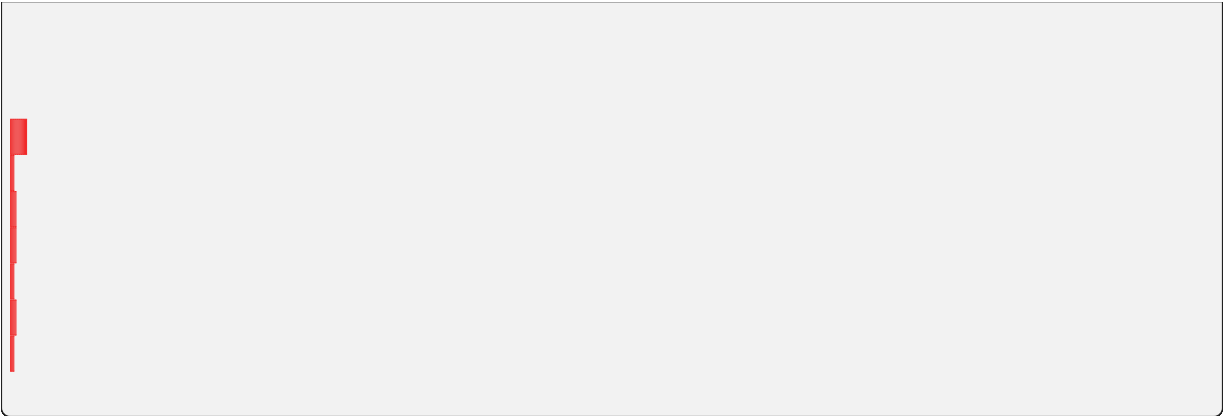




”

**Mutation Schema**

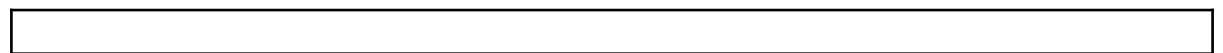




### 6.3.12 Basics of GraphQL



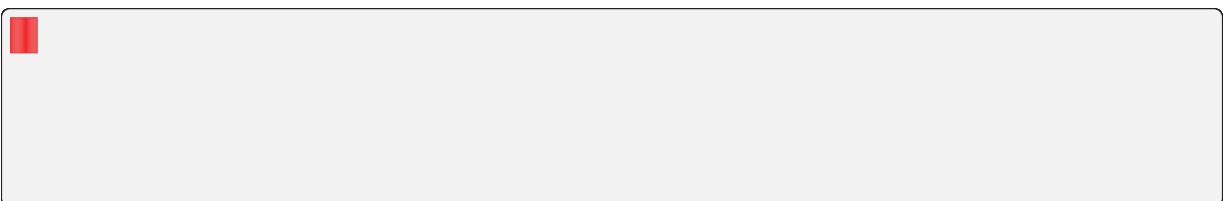
#### HTTP Request Convention



#### Field Naming Convention

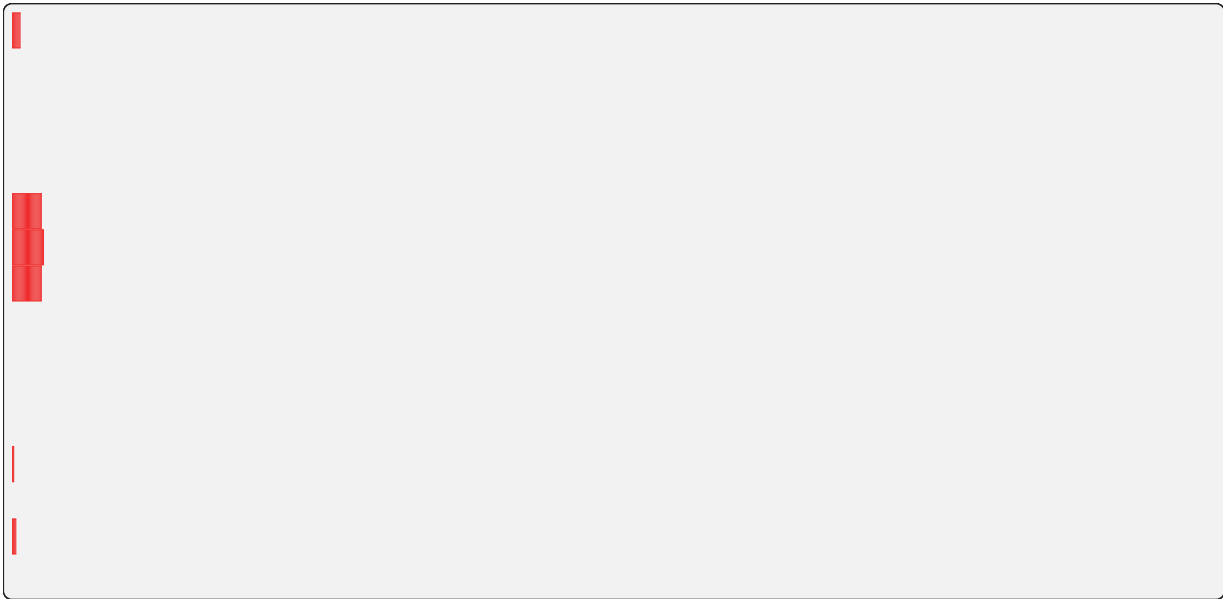
#### Common Object Types

“”





Pagination Convention



,

Custom Scalar Types

” ” ” ”

Authentication

Versioning

## 6.4 Backend.AI REST API Reference

## BACKEND.AI AGENT REFERENCE

### 7.1 RPC Interface for Kernel Management

### 7.2 Docker Backend

### 7.3 Kubernetes Backend

### 7.4 Accelerators (aka Compute Plugins)





## **BACKEND.AI STORAGE PROXY REFERENCE**

### **8.1 Storage Proxy Manager-facing API**

### **8.2 Storage Proxy Client-facing API**



## BACKEND.AI CLIENT SDK FOR PYTHON

---

”

---

## 9.1 Installation

### 9.1.1 Linux/macOS

---

---

### 9.1.2 Windows

### 9.1.3 Verification

“”

“”,

## 9.2 Client Configuration

,

## 9.3 Command Line Interface

### 9.3.1 Configuration

---

---

#### Session Mode

✓

✓

API Mode

Checking out the current configuration

9.3.2 Compute Sessions

Listing sessions

Option	Included Session Status

Option	Included Session Fields

### Running simple sessions

---

,

---

### Running sessions with accelerators

### Terminating or cancelling sessions

## 9.3.3 Container Applications

---

---

### Starting a session and connecting to its Jupyter Notebook

“” “”

### Accessing sessions via a web terminal

---

### Accessing sessions via native SSH/SFTP

“”

,



### 9.3.4 Storage Management

---

“”“”

#### Creating vfolders and managing them

“”:

File transfers and management

,

Running sessions with storages

“” “”

“”

Creating default files for kernels

“, ”

9.3.5 Advanced Code Execution

---

---

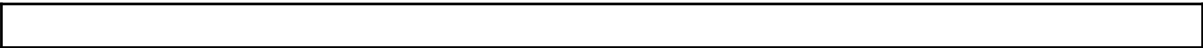
Running concurrent experiment sessions



“”

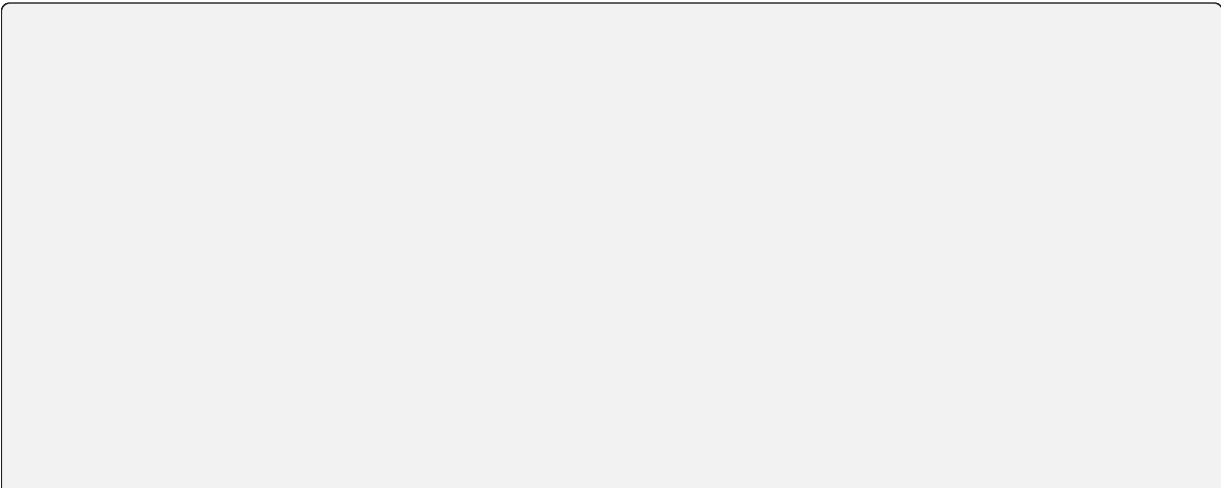
Expression	Interpretation

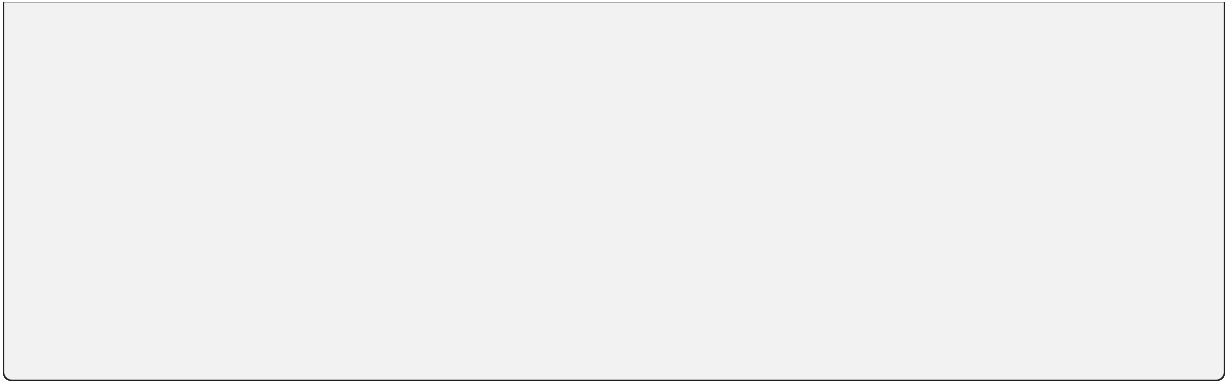
,



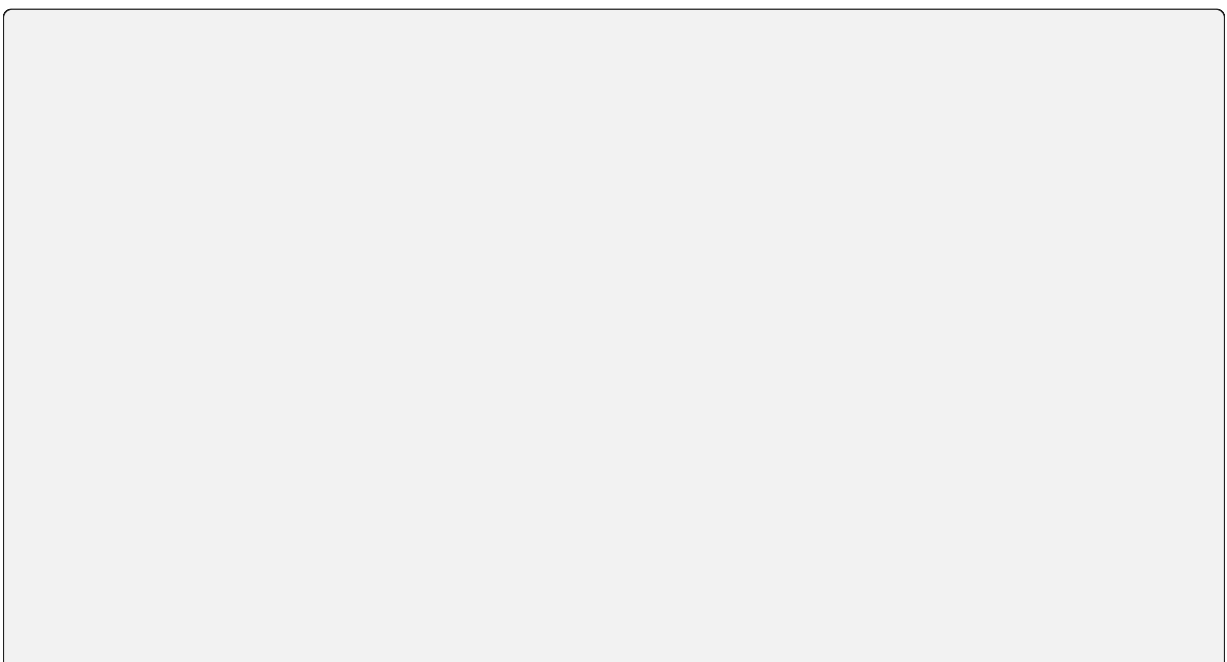
9.3.6 Session Templates

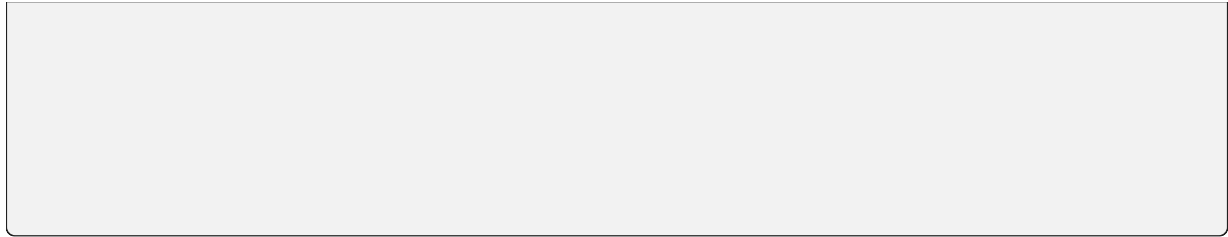
Creating and starting session template





#### Full syntax for task template





## 9.4 Developer Guides

### 9.4.1 Client Session

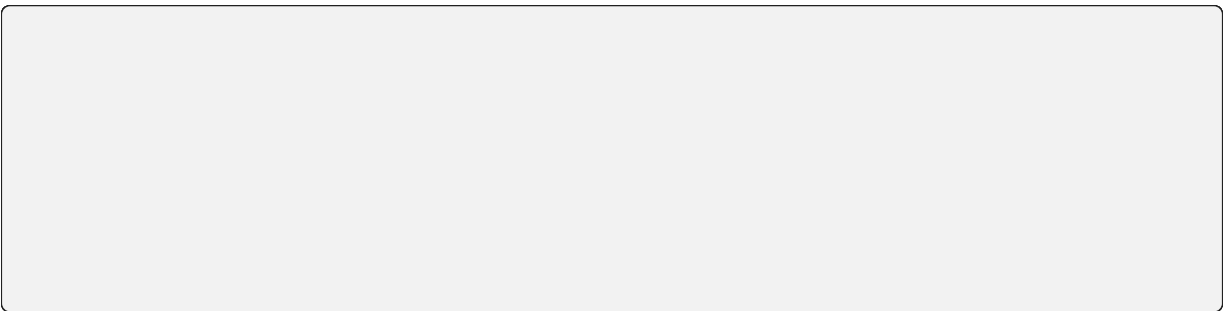
#### Client Session Objects

,

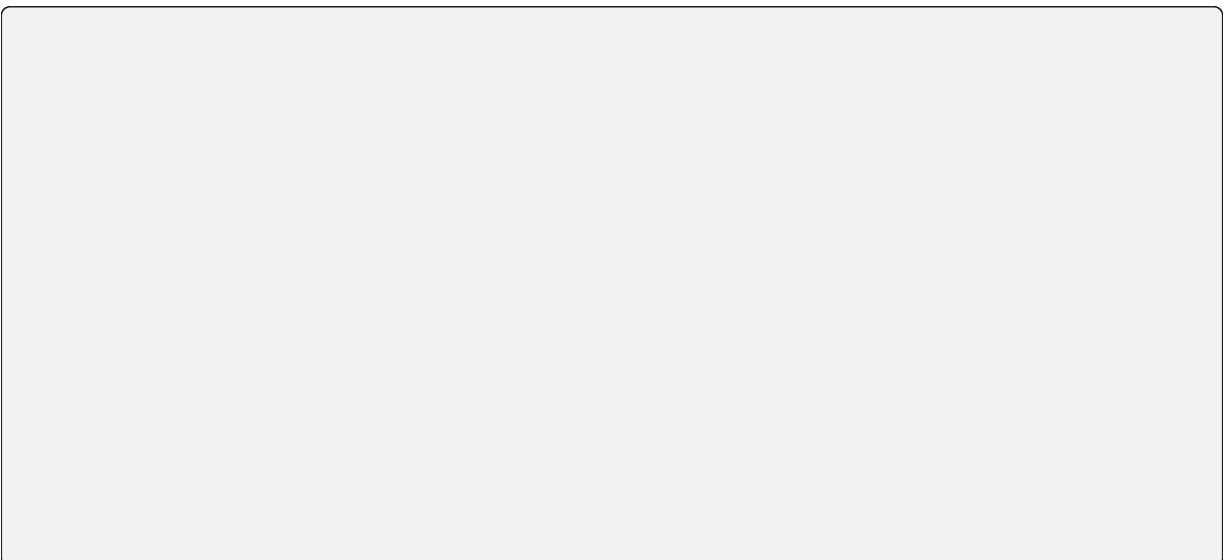
## 9.4.2 Examples

### Initialization of the API Client

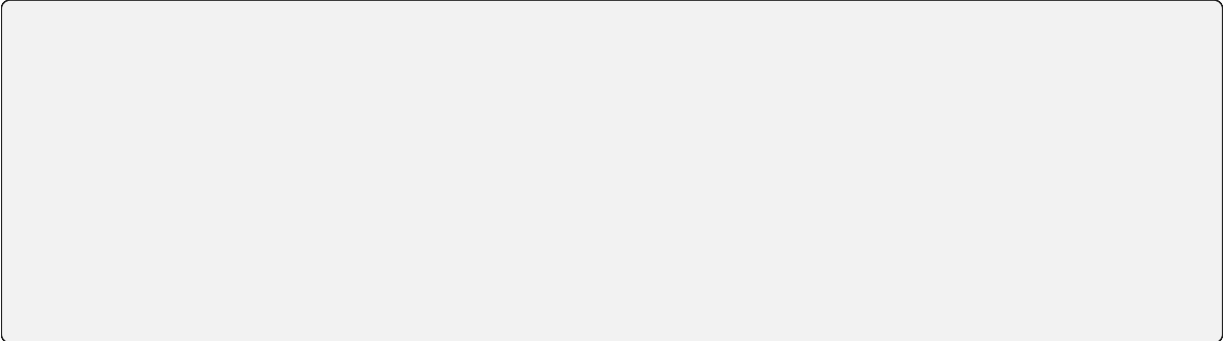
#### Implicit configuration from environment variables



#### Explicit configuration



### Asyncio-native API session

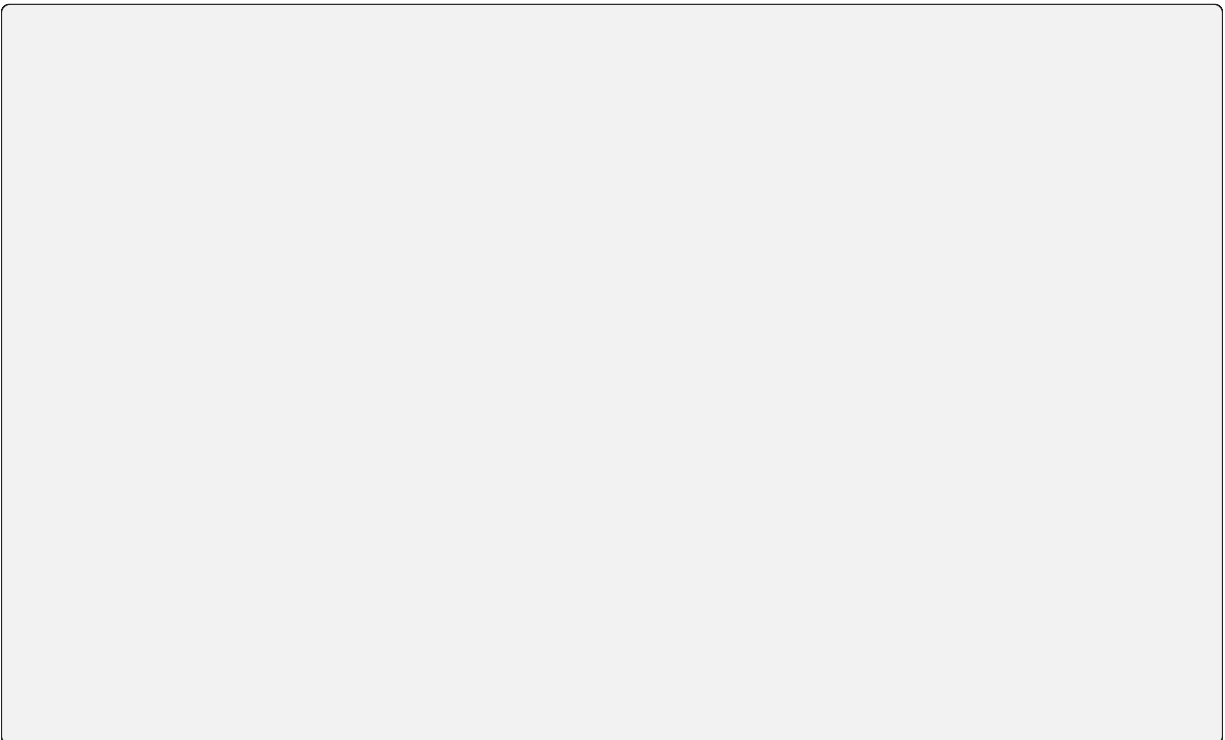


### Working with Compute Sessions

---

---

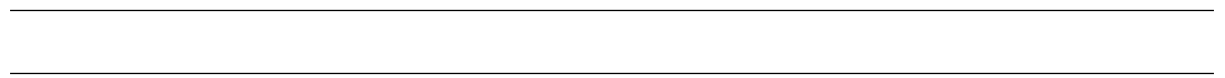
#### Listing currently running compute sessions



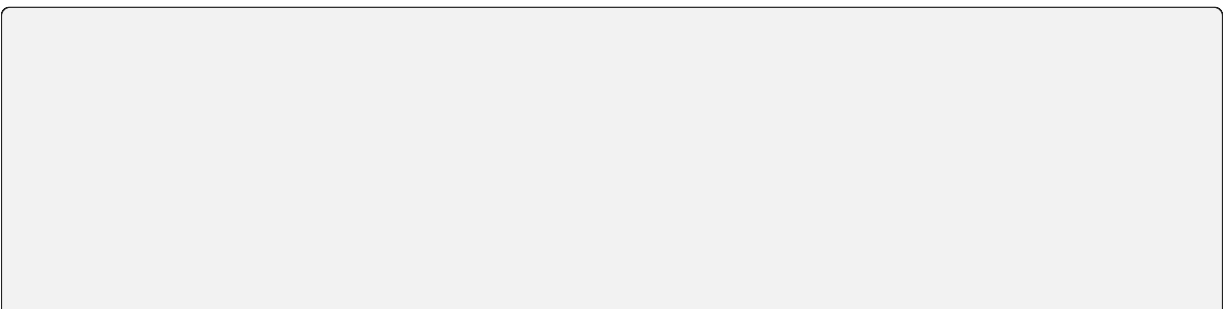
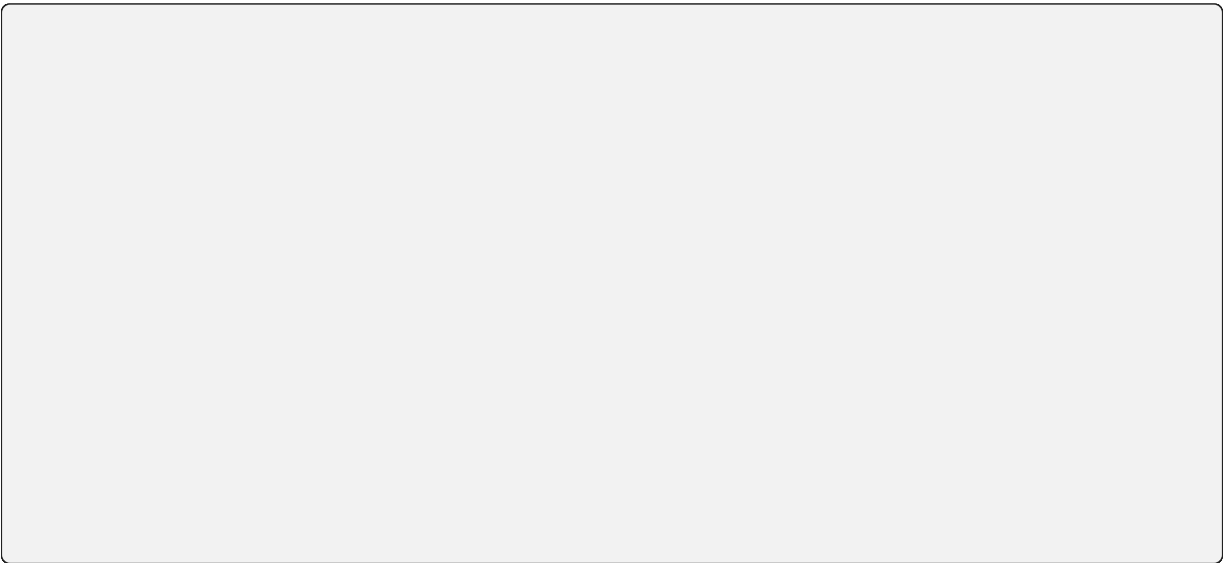
## Creating and destroying a compute session



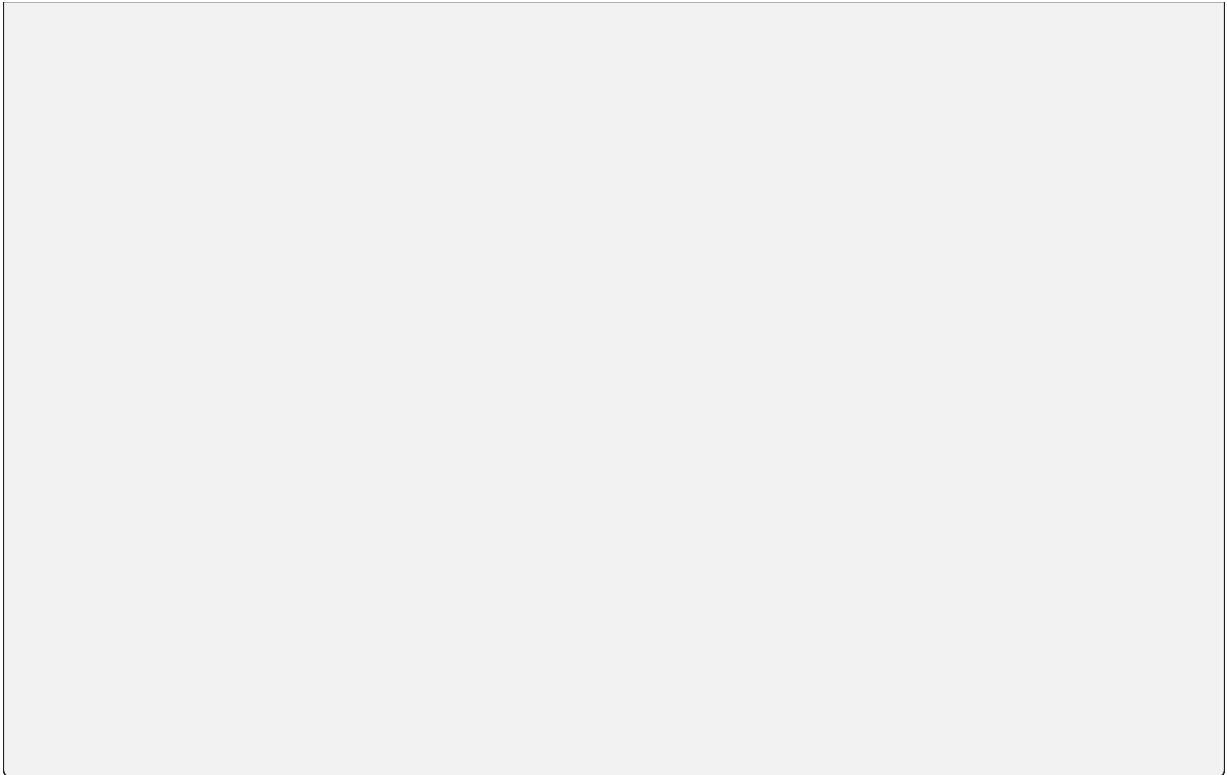
## Accessing Container Applications



## The API



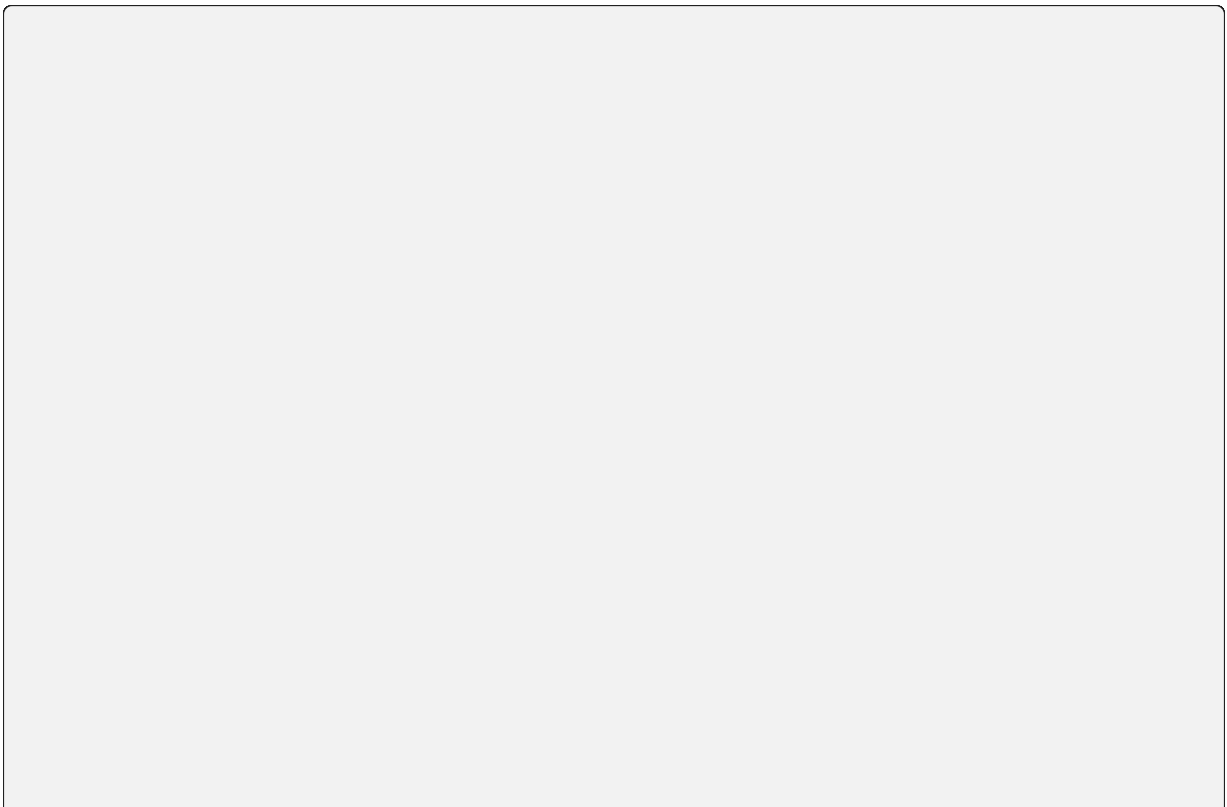




### Code Execution via API

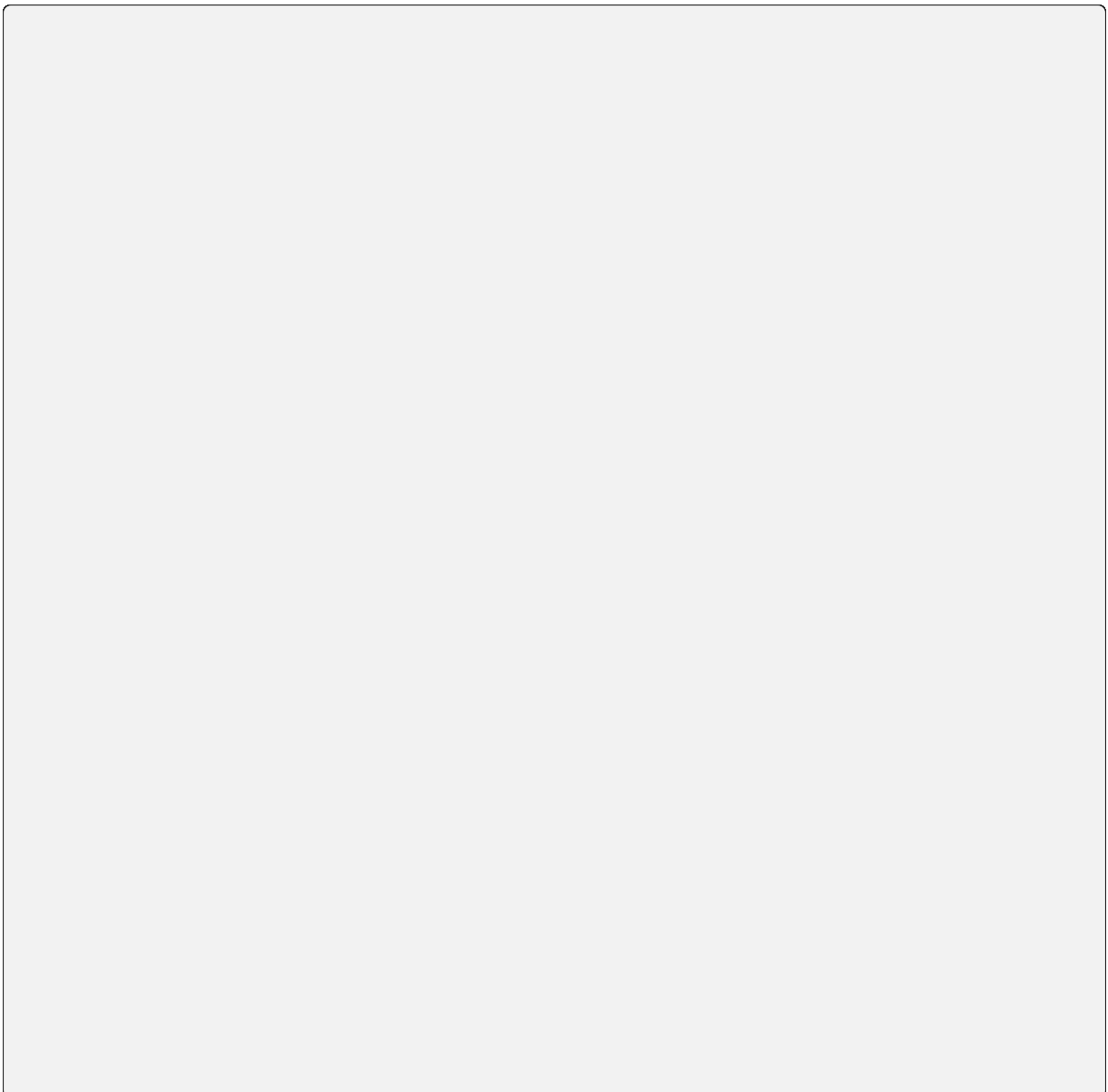
#### Synchronous mode

#### Snippet execution (query mode)

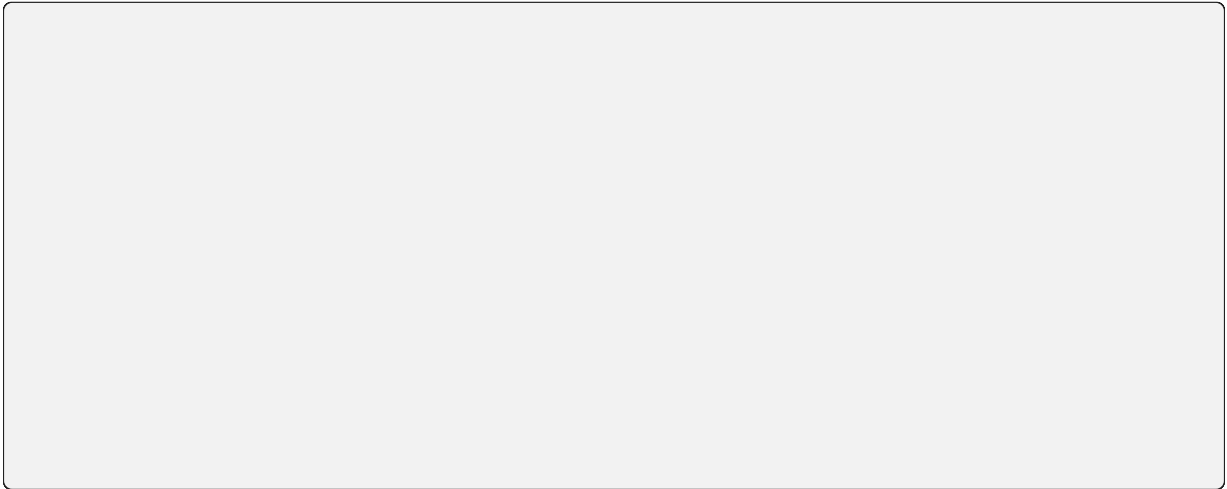




### Script execution (batch mode)



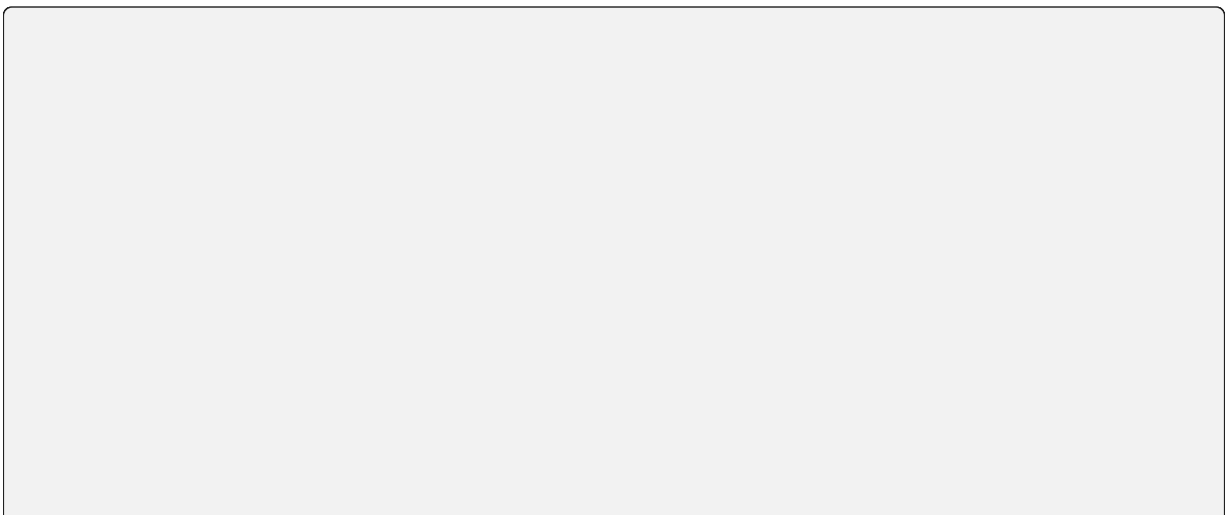
## Handling user inputs

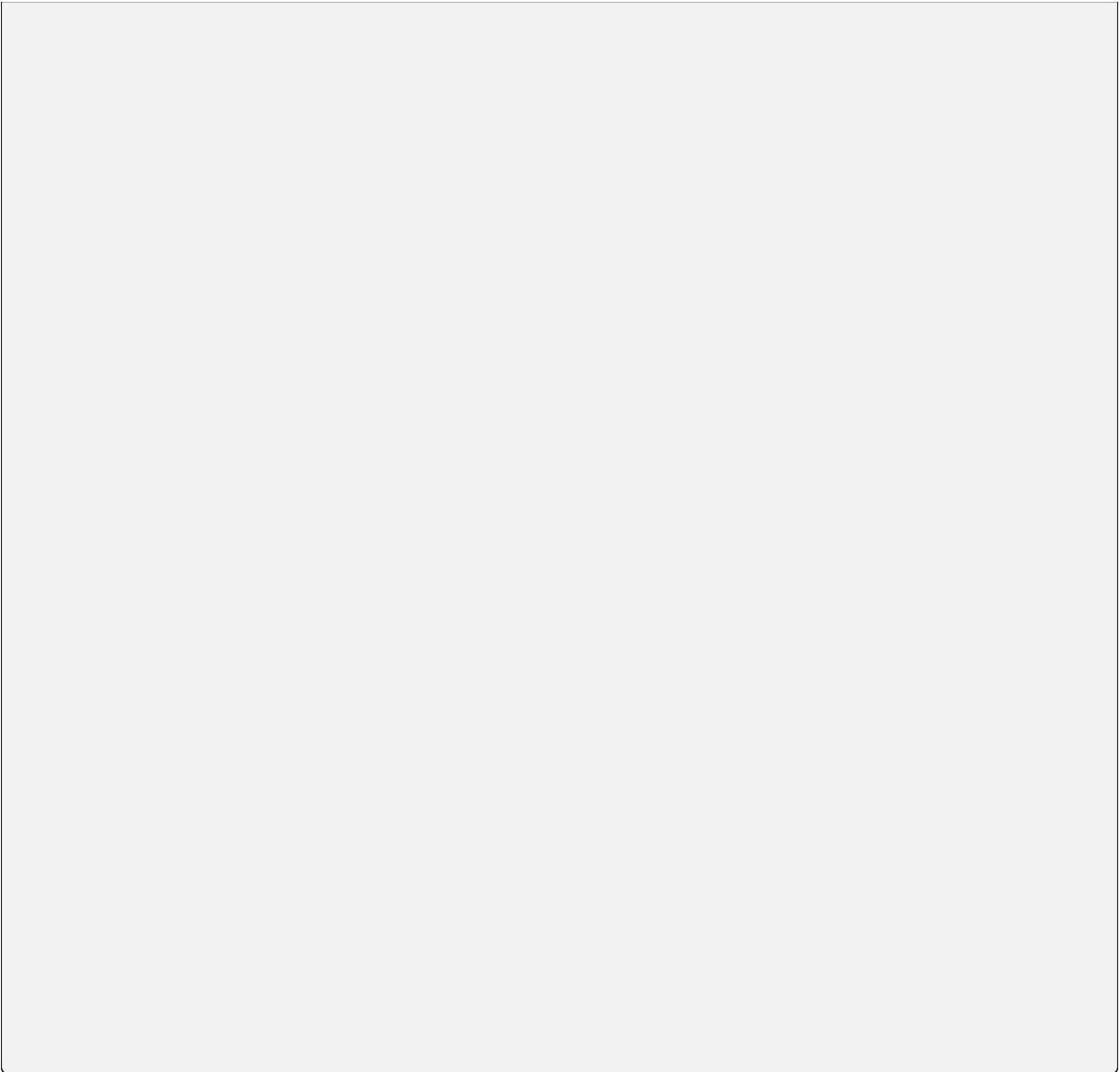


## Handling multi-media outputs



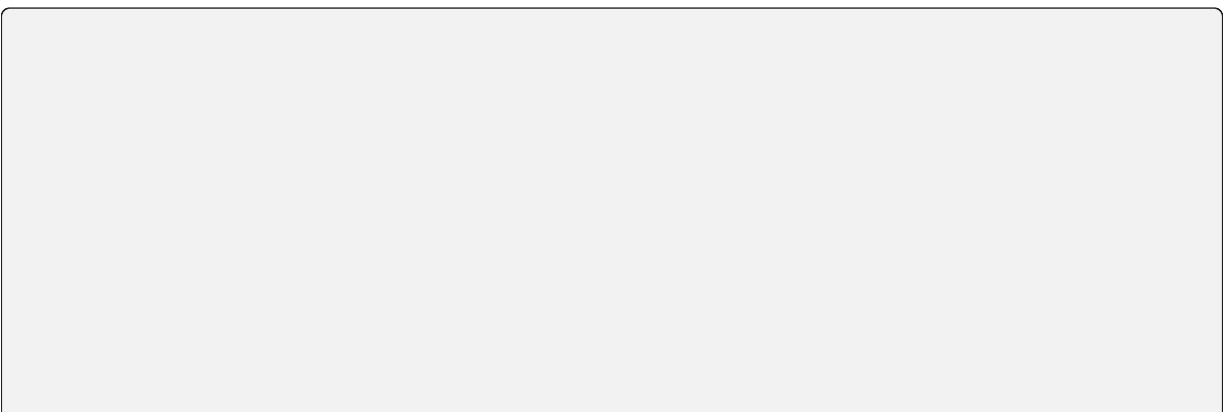
## Asynchronous mode





## Working with model service

### Starting model service



---

### Making request to model service endpoint

## 9.4.3 Testing

### Unit Tests

#### How to run

### Integration Tests

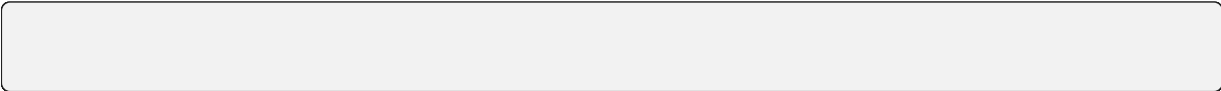
“”

### Prerequisite

---

---

How to run



## 9.5 High-level Function Reference

### 9.5.1 Admin Functions



## 9.5.2 Agent Functions

---

---

## 9.5.3 Auth Functions

,

,

,

## 9.5.4 Configuration