Extending oVirt UI with backup & recovery operation plugin - vProtect example

Marcin Kubacki
Chief Software Architect

September/2020
Agenda

- Getting started with plugin development
- vProtect overview
- vProtect UI Plugin overview
- Demo
Getting started with plugin development
Getting started with plugin development


- Install npm packages using command `yarn install` in the project directory.
Getting started with plugin development - overview

- **ui-extensions** directory with the **Dashboard** page (plugin).
- Build this project using command `yarn build` -> output in `dist` directory
- Ready plugins reside in `/usr/share/ovirt-engine/ui-plugins` directory in the oVirt Manager server

![Image of terminal output]

- Transforming Dashboard (dashboard* or ui-* files) plugin to Hello World (hello-world* files):
  - **hello-world-resources** directory - to be distributed
  - **hello-world.json** configuration file
  - **webpack.common.js** and **consts.js** configuration files - replace references to dashboard* with hello-world*
  - **hello-world.js** - entry component
Getting started with plugin development - hello-world.js
Getting started with plugin development - webpack config

- Modify dashboard entries to respective hello-world

```javascript
entry: {
  'plugin': [...commonModules, './src/plugin.js'],
  'dashboard': [...commonModules, './src/dashboard.js'],
  'hello-world': [...commonModules, './src/hello-world.js']
},

new HtmlWebpackPlugin({
  filename: 'plugin.html',
  template: 'static/html/plugin.template.ejs',
  extraParams: { gitinfo, rpminfo },
  inject: true,
  chunks: ['webpack-manifest', 'vendor', 'plugin']
}),

// new HtmlWebpackPlugin({
//  filename: 'dashboard.html',
//  template: 'static/html/dashboard.template.ejs',
//  extraParams: { gitinfo, rpminfo },
//  inject: true,
//  chunks: ['webpack-manifest', 'vendor', 'dashboard']
// })

new HtmlWebpackPlugin({
  filename: 'hello-world.html',
  template: 'static/html/hello-world.template.ejs',
  extraParams: { gitinfo, rpminfo },
  inject: true,
  chunks: ['webpack-manifest', 'vendor', 'hello-world']
})
```
Getting started with plugin development - oVirt menu item

- At integrations/places.js file comment out method responsible for adding Dashboard menu item.

```javascript
function addDashboardPlace() {
    // getPluginApi().addPrimaryMenuPlace(msg.dashboardTitle), dashboardPlaceToken, '{pluginBasePath}/dashboard.html', {
    // // place the menu item before existing ones
    // priority: -1,
    // // customize the prefix displayed in search bar
    // searchPrefix: 'Dashboard',
    // // make users land on this place by default
    // defaultPlace: true,
    // // make sure the menu item has the right icon
    // icon: 'fa-tachometer'
    // }
    getPluginApi().addPrimaryMenuPlace('Hello World', 'hello-world-place-token', '{pluginBasePath}/hello-world.html', {
        priority: -1,
        searchPrefix: 'Hello world',
        defaultPlace: true,
        icon: 'fa-tachometer'
    })
}
```

- Place token is a string that is displayed after „#” character in oVirt web engine URL.

- Using this route we will load our hello-world template, which is an react app with hello-world.js entry.
Getting started with plugin development - build

- yarn build ->

- It will generate the hello-world-extensions.json file inside. It will look like static/hello-world-extensions.json - place to pass config params to plugin

```json
{
  "name": "hello-world",
  "url": "plugin/hello-world/plugin.html",
  "resourcePath": "hello-world-resources",
  "lazyLoad": false,

  "config": {
    "useFakeData": false,
    "clusterUpgradePlaybook": "ovirt-cluster-upgrade"
  }
}
```
Getting started with plugin development - deploy

- Copy file and directory to oVirt Manager and it should be displayed
vProtect overview
vProtect overview

Backup solution for multiple virtualization platforms including oVirt-based, such as RHV or OLVM:

- Agent-less backups
- File-level restores
- Snapshot management
- Multiple backup providers supported
- Application-level backup mechanism
vProtect architecture
vProtect UI plugin overview
vProtect UI plugin overview

Technology stack of the project includes:
- React / Redux
- SASS
- PatternFly3 and Prime React components
- Yarn package manager
- Webpack
vProtect UI code structure

- Reusable components used throughout the project.
- Integrations with the oVirt web engine.
- Object models.
- Services for communication with the vProtect API.
- Redux store for application state management.
- Theme to customize style of components.
- Initial oVirt engine plugin template was written in JavaScript but we’re migrating it to TypeScript.

Github page: https://github.com/Storware/ovirt-engine-ui-vprotect-extensions
PatternFly3 and PrimeReact components

- oVirt web engine mainly uses PatternFly3 components
- we also used PrimeReact components
  - some components not implemented in PatternFly for React, i.e. DateTimePicker.
Communication workflow

- oVirt / RHV / OLVM
- vProtect UI Plugin
- vProtect Server
- vProtect Node

Click!

- vProtect API call
- Node fetches tasks and configuration from vProtect Server (vProtect API)

oVirt API - Java SDK
oVirt SDK for Java

- SDKs available for different languages
- You also can invoke APIs directly (HTTP) or with Ansible
- Java example:

```java
public Vm getVm(String id) {
    return conn.systemService().vmsService().get().send().vm();
}
```

- `pom.xml`:

```xml
<dependency>
    <groupId>org.ovirt.engine.api</groupId>
    <artifactId>sdk</artifactId>
    <version>4.2.5</version>
</dependency>
```
Demo
Thank you!

https://ovirt.org

https://www.openvirtualization.pro
https://storware.eu

users@ovirt.org

info@storware.eu

@ovirt

@OpenVirtPro  @Storware