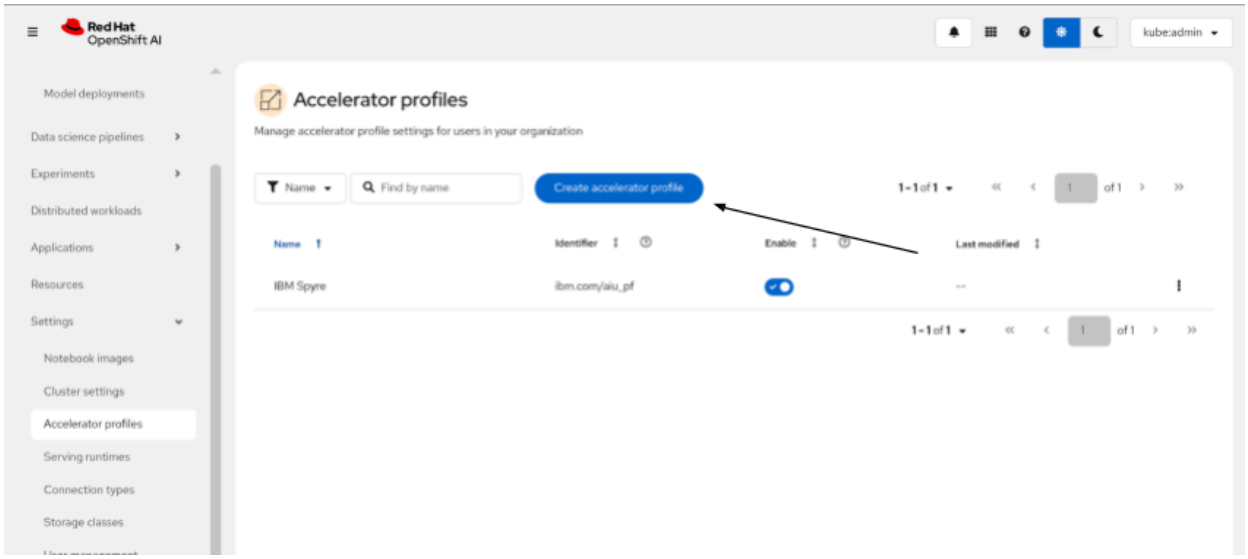
System information:

RHOAI 2.25
OCP 4.18.11
Spyre Operator v0.2.1

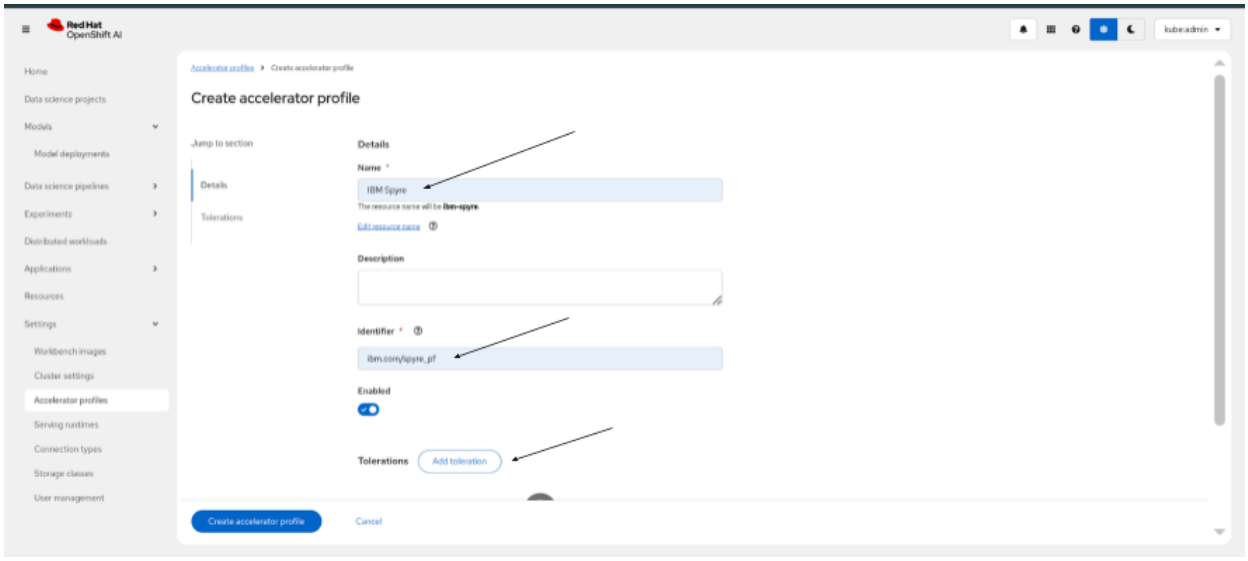
Setup Instructions

1. Install Spyre operator and OpenShift AI from OpenShift operator Hub
https://catalog.redhat.com/en/software/containers/ibm-aiu/spyre-operator/688a1121575e62c686a471d4
https://catalog.redhat.com/software/container-stacks/detail/63b85b573112fe5a95ee9a3a

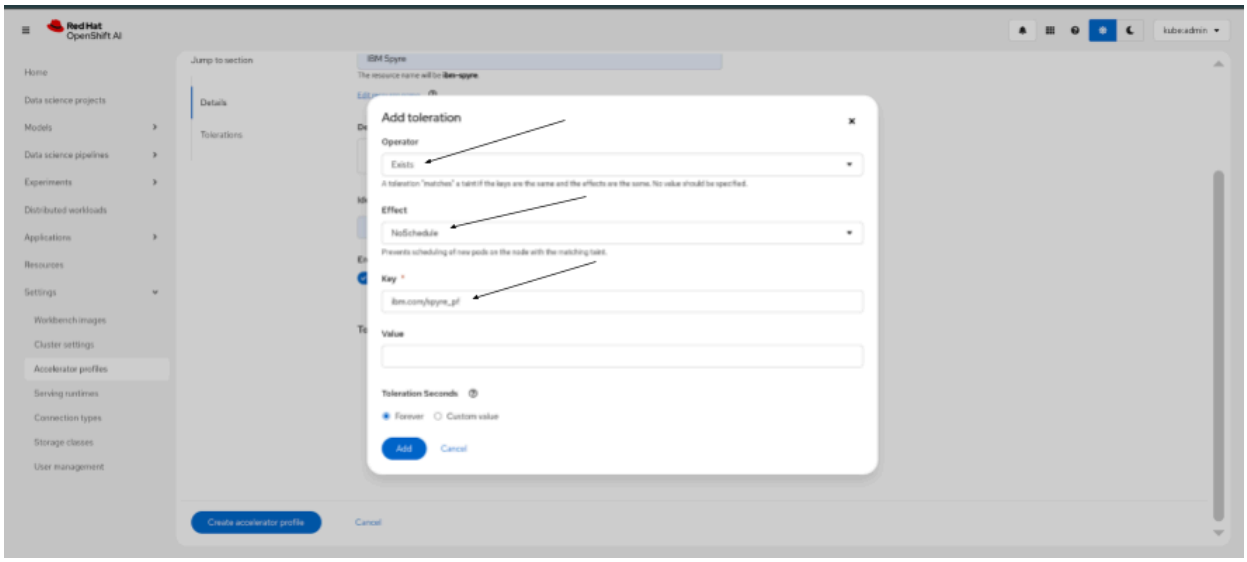
2. Create AcceleratorProfile in OpenShift AI Dashboard settings.



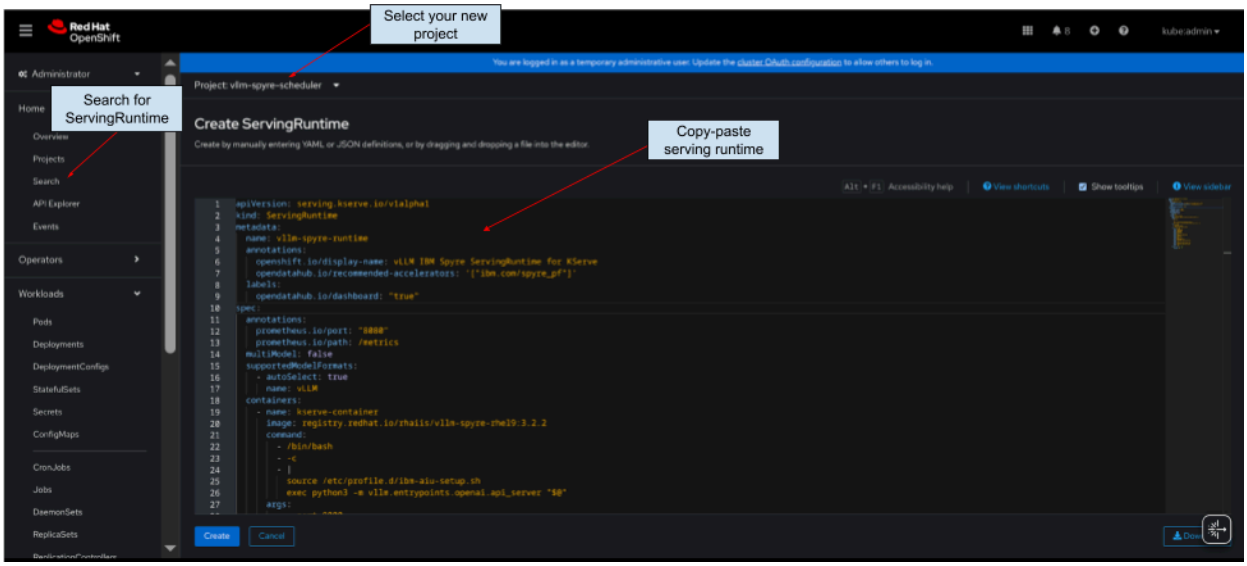
   a. Add Name and identifier



b. Add tolerations and click Add.



3. Add [custom ServingRuntime] in OpenShift.



4. Add registry pull-secret - Pull secrets are required to fetch images from registry.redhat.io where vLLM images and pre-built modelcar containers are kept e.g. oci://registry.redhat.io/rhelai1/modelcar-granite-3-1-8b-instruct:1.5 and oci://registry.redhat.io/rhelai1/modelcar-llama-3-1-8b-instruct:1.5 . This example uses a model from OCI registry https://docs.redhat.com/en/documentation/red_hat_ai_inference_server/3.0/html/validated_models/red_hat_ai_validated_models

Use the pull secrets provided to IBM for this step

   a. Obtain access to registry.redhat.io
   b. Login using podman. Enter your username and password

```
None
podman login registry.redhat.io
```

When you log into the registry, your credentials are stored in your ${XDG_RUNTIME_DIR}/containers/auth.json file. Those credentials are used automatically the next time you pull from that registry. Here is an example of that file:

```
Shell


{ "auths": { "https://registry.redhat.io": { "auth": "c2xmams6c2RmbGtq" } } }
```

c. Alternatively, you can create the authentication using this command. Which is a combination of your username and password.

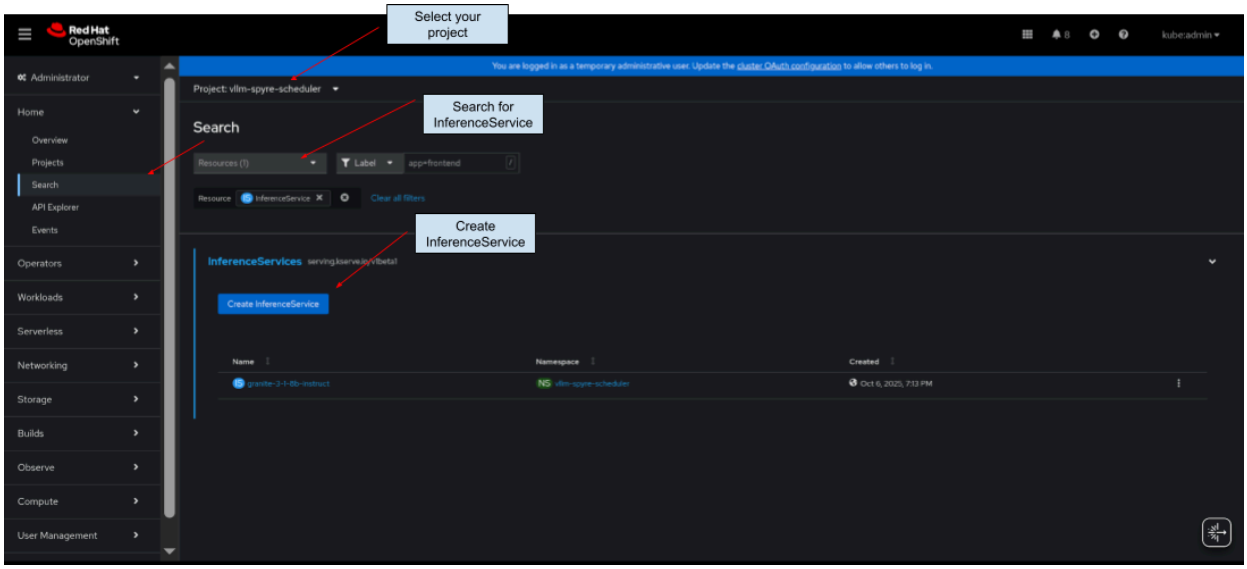
```
Shell
AUTH_STRING=$(echo -n "YOUR_USERNAME:YOUR_PASSWORD_HERE" | base64)
echo '{
    "auths": {
        "registry.redhat.io": {
            "auth": "'${AUTH_STRING}'"
        }
    }
}'
```

d. Alternatively apply from the terminal

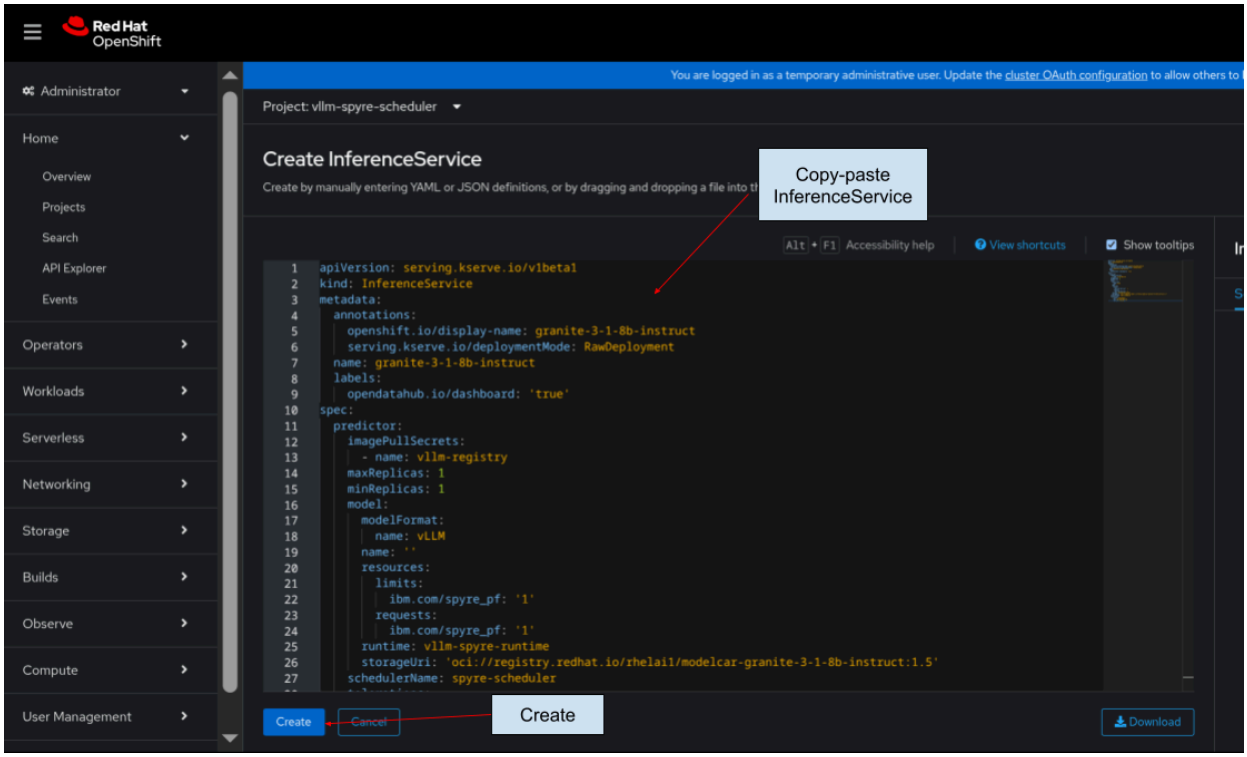
```
Shell
oc create secret docker-registry vllm-registry \
  --docker-server=registry.redhat.io \
  --docker-username=YOUR_USERNAME \
  --docker-password='YOUR_PASSWORD' \
  -n your-namespace
```

e. If working on a shared cluster, follow the link below to enable Registry Service Accounts for Shared Environments
https://access.redhat.com/articles/RegistryAuthentication

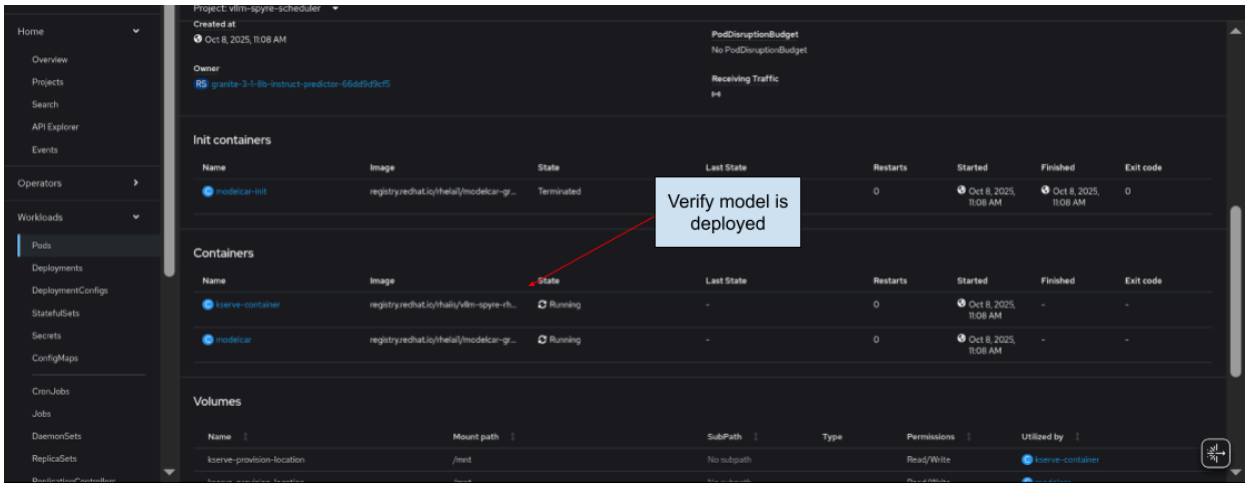
5. Create InferenceService - RawDeployment or InferenceService - Serverless in Openshift. Note that RHOAI Dashboard is missing the scheduler feature required for Spyre hardware profiles so we're using OpenShift for now.
    a. Search for InferenceService in your project and Create a new InferenceService



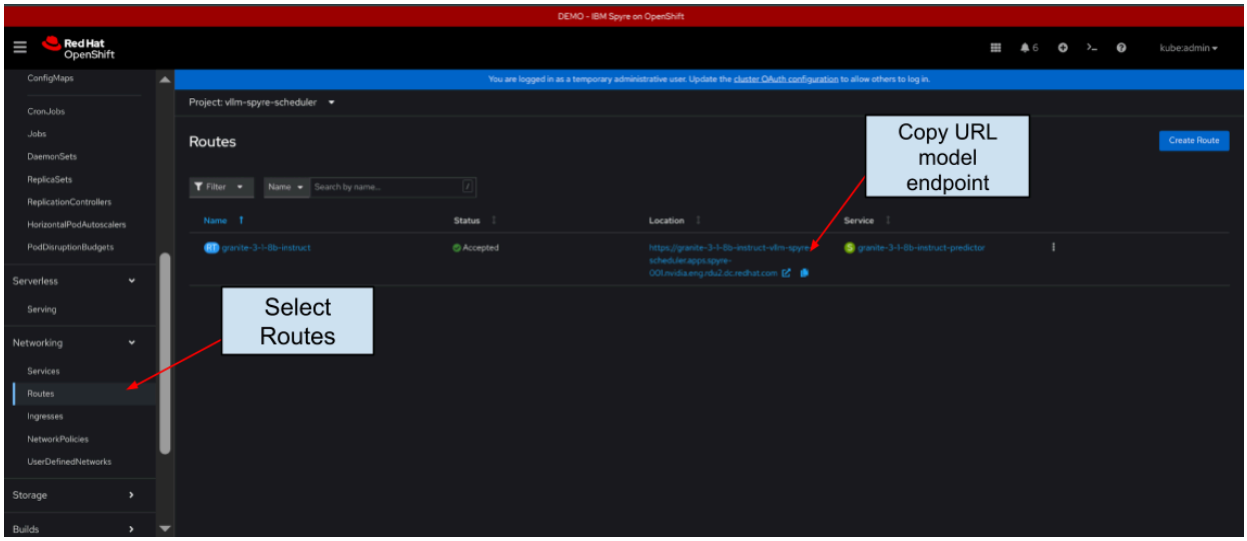
b. Copy-paste [InferenceService Yaml] into the editor and click “create”.



c. Verify that model is deployed



6. Model Inference
    a. Find the URL endpoint for model inference



    b. Send an inference request to the deployed model endpoint using any REST client of your choice.

```shell
curl -k
"https://granite-3-1-8b-instruct-vllm-spyre-scheduler.apps.spyre-001.nvidia.eng
.rdu2.dc.redhat.com/v1/completions" \
  -H "Content-Type: application/json" \
  -d '{"model":"granite-3-1-8b-instruct","prompt":"Write a short
poem.","temperature":0,"max_tokens":128}' \
| jq
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   897  100   798  100    99     38      4  0:00:24  0:00:20  0:00:04   180
{
  "id": "cmpl-4a2c14e12cf2401ca01a12aa03b28ebc",
  "object": "text_completion",
  "created": 1762888026,
  "model": "granite-3-1-8b-instruct",
  "choices": [
    {
      "index": 0,
      "text": "\n\nIn the quiet of the night, under the silver moon's
glow,\nStars twinkle like secrets, in the sky they sow.\nWhispers of the wind,
through the trees they weave,\nA symphony of silence, in the world we
believe.\n\nDreams take flight, on wings of the night,\nIn the canvas of
darkness, they ignite.\nTomorrow's promise, in the stars we see,\nIn the heart
of the night, hope is free.",
      "logprobs": null,
      "finish_reason": "stop",
      "stop_reason": null,
      "prompt_logprobs": null
    }
  ],
  "service_tier": null,
  "system_fingerprint": null,
  "usage": {
    "prompt_tokens": 6,
    "total_tokens": 120,
    "completion_tokens": 114,
    "prompt_tokens_details": null
  },
  "kv_transfer_params": null
}
```