**VVP w/ OOM deployment**

**IMPORTANT**
As of the Dublin release, the web components of VVP are deprecated and no longer maintained. The associated repositories in VVP have been locked, but they remain available for reference and historical purposes. OOM support has been removed in Dublin, but the Casablanca version still supports the Casablanca version of VVP.

**Installation**

**Deployment**

Installation is performed by OOM - see [http://onap.readthedocs.io/en/latest/submodules/oom.git/docs/oom_user_guide.html#user-guide-label](http://onap.readthedocs.io/en/latest/submodules/oom.git/docs/oom_user_guide.html#user-guide-label) for instructions. This guide assumes you are using a kubernetes cluster w/ rancher to deploy OOM.

In the instructions/examples below, I am using the following infrastructure:

- rancher node: 1.2.3.4
- kubernetes master node: 1.2.3.5
- kubernetes slave node 1: 1.2.3.6
- kubernetes slave node 0: 1.2.3.7

After Infrastructure is deployed, install VVP w/ OOM:

```bash
# clone oom repo
# update kubernetes/onap/values.yaml to enable/disable components.
# make sure VVP is enabled.

cd kubernetes
helm init --upgrade
helm serve &
helm repo add local http://127.0.0.1:8879
make all
helm install local/onap --name dev --namespace onap
```

**root@rancher:**

```bash
# kubectl get pods
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev-ci-uwsgi-688f759846-dcm5t</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>54m</td>
</tr>
<tr>
<td>dev-cms-uwsgi-564b5b5d86-mpgk1</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>54m</td>
</tr>
<tr>
<td>dev-em-645d6bf539-h9tdq</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>54m</td>
</tr>
<tr>
<td>dev-extend-haproxy-6597f77789-q15vn</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>54m</td>
</tr>
<tr>
<td>dev-imagescanner-54fbb84dfb8-6hx6</td>
<td>3/3</td>
<td>Running</td>
<td>2</td>
<td>54m</td>
</tr>
<tr>
<td>dev-int-haproxy-57945994cf-rj2pr</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>54m</td>
</tr>
<tr>
<td>dev-vvp-79b59c4f55-g84r2</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>54m</td>
</tr>
<tr>
<td>dev-vvp-gitlab-5f48c7cf6c-nmbrf</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>54m</td>
</tr>
<tr>
<td>dev-vvp-jenkins-69758c6dc8-cbm99</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>54m</td>
</tr>
<tr>
<td>dev-vvp-postgres-d9b9fd55-9vb4g</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>54m</td>
</tr>
<tr>
<td>dev-vvp-redis-7c7448c5f4-5is98</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>54m</td>
</tr>
</tbody>
</table>

**Post Deployment**

After deployment has been initiated, it will take ~10 minutes for the containers to come up and become functional.
To access the portal, you will need to login to the kubernetes dashboard and figure out what port is being used by the ext-haproxy pod mapped to internal port 80.

Once you have the port, access the portal by navigating in a web browser to http://1.2.3.5:31601.
To create an account

- Sign up like normal (email, pwd)

No activation email will be sent, you can find the activation link with the command

```
root@rancher:~# kubectl logs dev-em-645d6bf559-h9tdq
```

There will be a message with the activation link to navigate to in your web browser.

- login to your new account and create an engagement
- Add a public ssh key your profile

To login to the admin account

- From the sign up page, click the 'reset password' button
- use email admin@na.com
- Perform the same exercise as signing up, by browsing the engagement manager pod logs for the password reset link
- Paste the password reset link into your browser, and use the temp password from the logs
- Create a new password

Create an engagement
Once you create an engagement from the portal, a corresponding git repository will be created in the vvp-gitlab pod. To access the gitlab dashboard, you need to figure out what port is mapped for it. To do this, perform the same exercise as for the ext-haproxy pod, except instead now find the vvp-gitlab external port.

Navigate to http://1.2.3.5:<gitlab port> in your web browser. The root password is stored in the gitlab configmap and is base64 encoded. You should change the password as soon as possible.

Validation Scripts
Once heat templates are uploaded to gitlab, add those files to the checklist in the engagement manager portal.

After updating the checklist, jenkins pod will clone and run the validation-scripts against the heat templates. The results will be reported back to the jenkins log on the portal.

Once satisfied, move engagement to completed.
Known Issues

- If you are intending to only use VVP as an internal tool and not accessible publicly, you can mark the new git repo as "public" in the settings of the gitlab dashboard.
- Need to manually install validation-scripts python requirements. Clone validation scripts repo to jenkins pod after deployment, and run pip install -r requirements.txt.
- If VVP is available publicly but you wish to keep the heat templates stored in it private, you will need to modify the jenkins container file /usr/local /bin/ice-testengine to include the username/password for gitlab when cloning the heat templates from the gitlab instance.
- To clone the git repo from gitlab, you will only be able to use http.