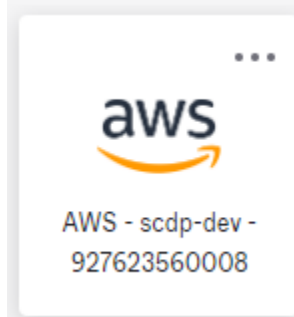




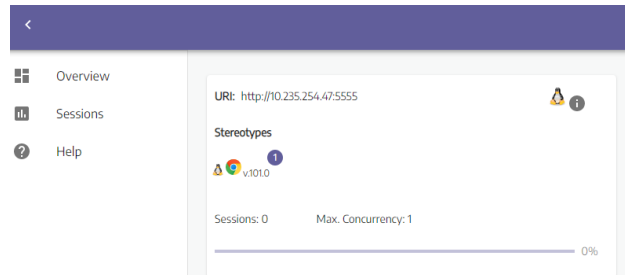


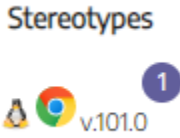
<p>The purpose of this document is to create a Selenium-grid on an EKS cluster. It will give you all the relevant commands but will not provide details on what these commands are actually doing. To understand these commands, I recommend doing a introduction to Kubernetes course</p> <p>The document will also not cover the AWS configuration side (i.e EKS and instance maintenance) as this has significant impact on other software that is hosted in pods</p> <p>You will need to have AWS role: <b>coupa-sso-dev-power-scdpdev</b></p>	
Access <b>AWS - scdp-dev - 927623560008</b> via the Coupa Portal	
Confirm the Region selected is <b>Ohio</b> (us-east-2)	
Launch Cloud Shell and wait for the terminal to be ready	
Type command <code>aws sts get-caller-identity</code> and verify you have a valid token	<pre>{   "UserId":     "ARO5P6VMHNEL47B6XE5Z:martin.day@coupa.com",   "Account": "927623560008",   "Arn":     "arn:aws:sts::927623560008:assumed-role/coupa-sso-dev-power-scdpdev/martin.day@coupa.com" }</pre>
Type command <code>kubectl config current-context</code>	<pre>arn:aws:eks:us-east-1:927623560008:cluster/eks-lt-dev</pre>
<p>Note if you do not get the instance name, type command  <code>kubectl config use-context</code>  <code>arn:aws:eks:us-east-1:927623560008:cluster/eks-lt-dev</code>  to access the Instance  <b>Kubectl</b> not available see Troubleshooting</p>	

<p>Confirm you have a folder called <b>sgrid</b>, type linux command</p> <p><code>ls -a</code></p> <p>If no folder is available see Troubleshooting</p>	<pre>.. . sgrid</pre>
<p>We need to upload all the files that create the grid framework. These are YAML files and are located in our SpecflowAutomation git repo</p>	<pre>hub-deploy-ingress.yaml hub-deployment.yaml hub-service.yaml hub-service-pubsub.yaml namespace.yaml node-chrome-deployment.yaml node-firefox-deployment.yaml</pre>
<p>Using AWS CloudShell upload feature (<i>In Actions</i>) add all these yaml files. By default uploaded files are saved to the user folder. Once uploaded, move all the files to the <b>sgrid</b> folder using linux command</p> <p><code>mv *.yaml sgrid</code></p> <p>You could always zip the files prior to uploading and upload the zip file</p>	<p>All the YAML files will be located in the sgrid folder</p>
<p>First we need to create a namespace to isolate all our grid's resources</p> <p>Type command</p> <p><code>kubectl get namespaces</code></p>	<pre>NAME                STATUS  AGE default             Active  2y34d jenkins             Active  2y33d kube-node-lease     Active  2y34d kube-public         Active  2y34d kube-system         Active  2y34d lltm               Active  686d monitoring          Active  2y34d zalenium           Active  2y34d</pre>
<p>Type command</p> <p><code>kubectl create -f sgrid/namespace.yaml</code></p>	<pre>namespace/selenium-grid created</pre>
<p>Type command</p> <p><code>kubectl get namespaces</code></p>	<pre>NAME                STATUS  AGE default             Active  2y34d jenkins             Active  2y33d kube-node-lease     Active  2y34d kube-public         Active  2y34d kube-system         Active  2y34d lltm               Active  686d monitoring          Active  2y34d selenium-grid       Active  29s zalenium           Active  2y34d</pre>
<p>Now that we have a unique namespace created, we can change our local config to use it and no other namespace.</p>	<p>Context</p> <pre>"arn:aws:eks:us-east-1:927623560008:cluster/eks-It-dev" modified.</pre>

Type command <code>kubectl config set-context --current --namespace=selenium-grid</code>	
We can then start to create our grid deployment Type command <code>kubectl create -f sgrid/hub-deployment.yaml</code>	deployment.apps/selenium-hub created
Confirm that the deployment pod is running Type command <code>Kubectl get pods</code>	NAME                                READY STATUS  RESTARTS  AGE selenium-hub-79b667c579-n9qps  1/1 <b>Running</b> 0          69s
Next we need to create the grid services Type command <code>kubectl create -f sgrid/hub-service.yaml</code>	service/selenium-srv created
And then we need to publish service ports that are used by the grid Type command <code>kubectl create -f sgrid/hub-service-pubsub.yaml</code>	service/selenium-hub-pubsub created
Confirm that both new services are available. Type command <code>kubectl get services</code>	See row below
NAME                                TYPE             CLUSTER-IP      EXTERNAL-IP  PORT(S)              AGE <b>selenium-hub-pubsub</b> ClusterIP  172.20.248.177  <none>      4443/TCP,4442/TCP  62s <b>selenium-srv</b> NodePort   172.20.227.92   <none>      4444:30001/TCP     3m9s	
Now that we have a deployed grid with services, we can add browsers. First we will add <b>chrome</b> pod Type command <code>kubectl create -f sgrid/node-chrome-deployment.yaml</code>	deployment.apps/selenium-node-chrome created
Confirm that the <b>chrome</b> node pods are running. There should be <b>10</b> of them Type command <code>Kubectl get pods</code>	NAME                                READY STATUS  RESTARTS  AGE selenium-hub-79b667c579-n9qps  1/1  Running  0          10m selenium-node-chrome-7fd9966cc6-2h88g  1/1  Running  0          19s selenium-node-chrome-7fd9966cc6-4pttg

	1/1    Running   0        19s selenium-node-chrome-7fd9966cc6-dlkj7 1/1    Running   0        19s selenium-node-chrome-7fd9966cc6-hhpc f 1/1    Running   0        19s selenium-node-chrome-7fd9966cc6-jnxz 7 1/1    Running   0        19s selenium-node-chrome-7fd9966cc6-lhzvz 0/1    Pending   0        19s selenium-node-chrome-7fd9966cc6-lktb9 1/1    Running   0        19s selenium-node-chrome-7fd9966cc6-r6rzh 0/1    Pending   0        19s selenium-node-chrome-7fd9966cc6-rxh7 n 1/1    Running   0        19s selenium-node-chrome-7fd9966cc6-xkx7 2 1/1    Running   0        19s
Optional: we can add firefox as well Type command <b>kubectl create -f</b> <b>sgrid/node-firefox-deployment.yaml</b> And then confirm firefox node pods are running. There will 5 by default Type command <b>Kubectl get pods</b> Note if you see status <b>pending</b> , keep using command <b>Kubectl get pods</b> until they change to running	selenium-node-firefox-745c88c5f6-h82q5 0/1    Pending   0        28s selenium-node-firefox-745c88c5f6-nlxpj 0/1    Pending   0        28s selenium-node-firefox-745c88c5f6-sjvbb 0/1    Pending   0        28s selenium-node-firefox-745c88c5f6-w9g2 q 0/1    Pending   0        28s selenium-node-firefox-745c88c5f6-xlxfr 0/1    Pending   0        28s
Confirm we now have all deployments available. Type command <b>kubectl get deployments</b>	See row below
<b>NAME</b> <b>READY</b> <b>UP-TO-DATE</b> <b>AVAILABLE</b> <b>AGE</b> selenium-hub                1/1       1               1               16m selenium-node-chrome      10/10   10               10               6m20s selenium-node-firefox      5/5       5               5               4m8s	
Next we need to make the pods IP address available to our network Type command <b>kubectl expose deployment</b> <b>selenium-hub --type=LoadBalancer</b> <b>--name=selenium-hub</b>	<b>service/selenium-hub exposed</b>
Confirm that the exposed deployment is available.	See row below

Type command kubectl get services	As this is hosted on AWS you will see an AWS address under the EXTERNAL-IP column																								
<table><tr><th>NAME</th><th>TYPE</th><th>CLUSTER-IP</th><th>EXTERNAL-IP</th></tr><tr><td>selenium-hub</td><td>LoadBalancer</td><td>172.20.248.8</td><td>a0f54103221dc456da52baa47c1d5458-1249501187.us-east-1.elb.amazonaws.com</td></tr><tr><td>selenium-hub-pubsub</td><td>ClusterIP</td><td>172.20.248.177</td><td>&lt;none&gt;</td></tr><tr><td>4443/TCP,4442/TCP</td><td></td><td>18m</td><td></td></tr><tr><td>selenium-srv</td><td>NodePort</td><td>172.20.227.92</td><td>&lt;none&gt;</td></tr><tr><td>4444:30001/TCP</td><td></td><td>20m</td><td></td></tr></table>		NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	selenium-hub	LoadBalancer	172.20.248.8	a0f54103221dc456da52baa47c1d5458-1249501187.us-east-1.elb.amazonaws.com	selenium-hub-pubsub	ClusterIP	172.20.248.177	<none>	4443/TCP,4442/TCP		18m		selenium-srv	NodePort	172.20.227.92	<none>	4444:30001/TCP		20m	
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP																						
selenium-hub	LoadBalancer	172.20.248.8	a0f54103221dc456da52baa47c1d5458-1249501187.us-east-1.elb.amazonaws.com																						
selenium-hub-pubsub	ClusterIP	172.20.248.177	<none>																						
4443/TCP,4442/TCP		18m																							
selenium-srv	NodePort	172.20.227.92	<none>																						
4444:30001/TCP		20m																							
Next we need to make the pods available to our network Type command kubectl create -f sgrid/hub-deploy-ingress.yaml	ingress.networking.k8s.io/selenium-ingress created																								
Confirm that the ingress is available Type command Kubectl get ingress	See row below																								
<table><tr><th>NAME</th><th>HOSTS</th><th>ADDRESS</th><th>PORTS</th><th>AGE</th></tr><tr><td>selenium-ingress</td><td>my-selenium-grid.com</td><td>ab866e556790911ea8ab302807db7995-c9e51ea5083a9068.elb.us-east-1.amazonaws.com</td><td>80</td><td>71s</td></tr></table>		NAME	HOSTS	ADDRESS	PORTS	AGE	selenium-ingress	my-selenium-grid.com	ab866e556790911ea8ab302807db7995-c9e51ea5083a9068.elb.us-east-1.amazonaws.com	80	71s														
NAME	HOSTS	ADDRESS	PORTS	AGE																					
selenium-ingress	my-selenium-grid.com	ab866e556790911ea8ab302807db7995-c9e51ea5083a9068.elb.us-east-1.amazonaws.com	80	71s																					
The grid is now completed and we need to confirm it is working (and accessible on the internet) Type command kubectl get services Copy the long AWS address under the External IP column for LoadBalancer type Open a web-browser and use url https://www.nslookup.io/ Paste the AWS address into the Domain Name and press find DNS records Make a note of any of the IP address listed	<div><h2>A records</h2><p>IPv4 address</p><div><div>&gt;</div><div> 52.4.92.0</div></div><div><div>&gt;</div><div> 3.221.154.34</div></div><div><div>&gt;</div><div> 52.3.84.166</div></div></div>																								
Open a web browser and using one of the previous IPv4 addresses type URL http://<IP address>:4444/ui/index.html For example http://52.3.84.166:4444/ui/index.html # Selenium-grid will be available in the browser																									

and exposed to the internet	
<p>Note the Chrome browser icon has a version number</p> <p>For example this is Chrome version 101</p> <p>This is obtained from the docker image located in the <b>node-chrome-deployment.yaml</b></p> <p>To update this image will be covered in the Selenium-grid read me</p>	
<p>To confirm that your new framework is healthy, type command <b>clear</b></p> <p>Followed by command <b>KubectI get all</b></p> <p>Confirm all pods are running and Ready column is populated with 1</p>	<p>Clear will refresh the screen</p> <p>See row below</p>
<pre> NAME                                READY  STATUS   RESTARTS  AGE pod/selenium-hub-79b667c579-n9qps    1/1    Running  0          52m pod/selenium-node-chrome-7fd9966cc6-2h88g  1/1    Running  0          42m pod/selenium-node-chrome-7fd9966cc6-4pttg  1/1    Running  0          42m pod/selenium-node-chrome-7fd9966cc6-dlkj7  1/1    Running  0          42m pod/selenium-node-chrome-7fd9966cc6-hhpcf  1/1    Running  0          42m pod/selenium-node-chrome-7fd9966cc6-jnxz7  1/1    Running  0          42m pod/selenium-node-chrome-7fd9966cc6-lhzvz  1/1    Running  0          42m pod/selenium-node-chrome-7fd9966cc6-lktb9  1/1    Running  0          42m pod/selenium-node-chrome-7fd9966cc6-r6rzh  1/1    Running  0          42m pod/selenium-node-chrome-7fd9966cc6-rxh7n  1/1    Running  0          42m pod/selenium-node-chrome-7fd9966cc6-xkx72  1/1    Running  0          42m </pre> <pre> NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP PORT(S)                            AGE service/selenium-hub                LoadBalancer        172.20.248.8 a0f54103221dc456da52baa47c1d5458-1249501187.us-east-1.elb.amazonaws.com 4444:31995/TCP,4443:32754/TCP,4442:32138/TCP  29m service/selenium-hub-pubsub          ClusterIP             172.20.248.177  &lt;none&gt; 4443/TCP,4442/TCP                    46m service/selenium-srv                NodePort              172.20.227.92  &lt;none&gt; 4444:30001/TCP                       49m </pre> <pre> NAME                                READY  UP-TO-DATE  AVAILABLE  AGE deployment.apps/selenium-hub         1/1    1            1          52m deployment.apps/selenium-node-chrome 10/10  10          10         42m </pre> <pre> NAME                                DESIRED  CURRENT  READY  AGE replicaset.apps/selenium-hub-79b667c579  1        1        1      52m </pre>	

```
replicaset.apps/selenium-node-chrome-7fd9966cc6 10 10 10 42m  
[cloudshell-user@ip-10-0-55-103 ~]$
```

## Troubleshooting

If you see

`kubectl not found`

follow this instructions on this link

<https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html>

And then to verify it is installed

`kubectl version --short --client`

When writing this guide I was using **V22**

If you see

The connection to the server localhost:8080 was refused - did you specify the right host or port?

Type command

`kubectl config get-contexts`

If the above command returns empty list, then you config file is missing, type command

`aws eks update-kubeconfig --name eks-lt-dev --region us-east-1`

Then repeat command

`kubectl config get-contexts`

Which will return you

`CURRENT NAME`

`CLUSTER`

`AUTHINFO`

`NAMESPACE`

`* arn:aws:eks:us-east-1:927623560008:cluster/eks-lt-dev`

`arn:aws:eks:us-east-1:927623560008:cluster/eks-lt-dev`

`arn:aws:eks:us-east-1:927623560008:cluster/eks-lt-dev`

No sgrid folder, type linux command

`mkdir sgrid`