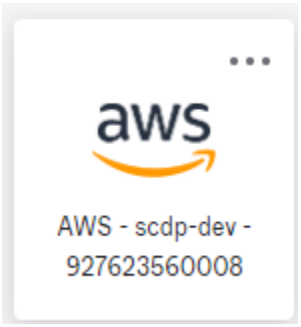
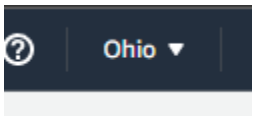
The purpose of this document is to create a Selenium-grid on an EKS cluster.
It will give you all the relevant commands but will not provide details on what these commands are actually doing. To understand these commands, I recommend doing a introduction to Kubernetes course
The document will also not cover the AWS configuration side (i.e EKS and instance maintenance) as this has significant impact on other software that is hosted in pods
You will need to have AWS role**: coupa-sso-dev-power-scdpdev**

Access **AWS - scdp-dev - 927623560008** via the Coupa Portal	
Confirm the Region selected is **Ohio** (us-east-2)	
Launch Cloud Shell and wait for the terminal to be ready	

Type command
aws sts get-caller-identity
and verify you have a valid token
	{
	"UserId": "AROA5P6VMHNEL47B6XE5Z:martin.day@coupa.com",
	"Account": "927623560008",
	"Arn": "arn:aws:sts::927623560008:assumed-role/coupa-sso-dev-power-scdpdev/martin.day@coupa.com"
	}

Type command kubectl config current-context	arn:aws:eks:us-east-1:927623560008:cluster /eks-lt-dev

Note if you do not get the instance name, type command
kubectl config use-context arn:aws:eks:us-east-1:927623560008:cluster/eks-lt-dev
to access the Instance
kubectl or cluster not available see **Troubleshooting**

Confirm you have a folder called **sgrid**, type linux command ls -a If no folder is available see Troubleshooting	.. . sgrid
We need to upload all the files that create the grid framework. These are YAML files and are located in our SpecflowAutomation git repo	hub-deploy-ingress.yaml hub-deployment.yaml hub-service.yaml hub-service-pubsub.yaml namespace.yaml node-chrome-deployment.yaml node-firefox-deployment.yaml
Using AWS CloudShell upload feature (*In Actions*) add all these yaml files. By default uploaded files are saved to the user folder. Since AWS CloudShell allows one file upload at a time. Instead of uploading individual files, before uploading copy all yaml files into a folder, name the folder YAML_Files and create a zip file for this folder Upload Above zip file to aws Once uploaded, follow next command to unzip the folder, move all yaml files to the **sgrid** folder and remove uploaded zip file/extracted folder from user home folder unzip YAML_Files.zip && mv YAML_Files/* sgrid && rm -d YAML_Files && rm YAML_Files.zip	All the YAML files will be located in the sgrid folder
First we need to create a namespace to isolate all our grid's resources Type command kubectl get namespaces	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
Type command kubectl create -f sgrid/namespace.yaml	namespace/selenium-grid created
Type command kubectl get namespaces	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
Now that we have a unique namespace created, we can change our local config to use it and no other namespace. Type command kubectl config set-context --current --namespace=selenium-grid	Context "arn:aws:eks:us-east-1:927623560008:cluster/eks-lt-dev" modified.
We can then start to create our grid deployment Type command kubectl create -f sgrid/hub-deployment.yaml	deployment.apps/selenium-hub created
Confirm that the deployment pod is running Type command kubectl get pods	NAME READY STATUS RESTARTS AGE selenium-hub-79b667c579-n9qps 1/1 **Running** 0 69s
Next we need to create the grid services Type command kubectl create -f sgrid/hub-service.yaml See troubleshooting for any known issues	service/selenium-srv created
And then we need to publish service ports that are used by the grid Type command kubectl create -f sgrid/hub-service-pubsub.yaml	service/selenium-hub-pubsub created
Confirm that both new services are available. Type command kubectl get services	See row below
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE **selenium-hub-pubsub** ClusterIP 172.20.248.177 <none> 4443/TCP,4442/TCP 62s **selenium-srv** 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 **chrome** pod Type command kubectl create -f sgrid/node-chrome-deployment.yaml	deployment.apps/selenium-node-chrome created
Confirm that the **chrome** node pods are running. There should be **10** of them Type command kubectl get pods	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-hhpcf 1/1 Running 0 19s selenium-node-chrome-7fd9966cc6-jnxz7 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-rxh7n 1/1 Running 0 19s selenium-node-chrome-7fd9966cc6-xkx72 1/1 Running 0 19s
Optional: we can add firefox as well Type command kubectl create -f sgrid/node-firefox-deployment.yaml And then confirm firefox node pods are running. There will 5 by default Type command kubectl get pods Note if you see status pending, keep using command kubectl get pods 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-w9g2q 0/1 Pending 0 28s selenium-node-firefox-745c88c5f6-xlxfr 0/1 Pending 0 28s
Confirm we now have all deployments available. Type command kubectl get deployments	See row below

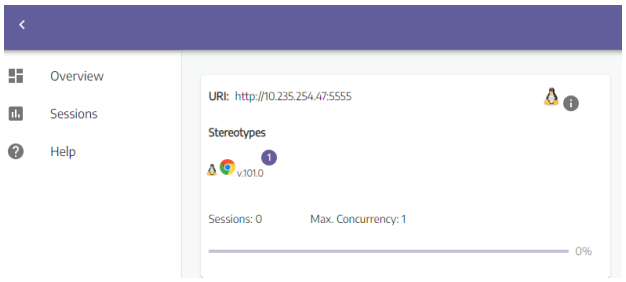
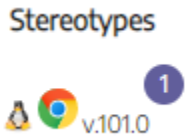
NAME READY UP-TO-DATE AVAILABLE AGE selenium-hub 1/1 1 1 16m selenium-node-chrome 10/10 10 10 6m20s selenium-node-firefox 5/5 5 5 4m8s

Next we need to make the pods IP address available to our network Type command kubectl expose deployment selenium-hub --type=LoadBalancer --name=selenium-hub	service/selenium-hub exposed
Confirm that the exposed deployment is available. Type command kubectl get services	See row below As this is hosted on AWS you will see an AWS address under the EXTERNAL-IP column

NAME TYPE CLUSTER-IP EXTERNAL-IP **selenium-hub LoadBalancer 172.20.248.8 a0f54103221dc456da52baa47c1d5458-1249501187.us-east-1.elb.amazonaws.com 4444:31995/TCP,4443:32754/TCP,4442:32138/TCP 62s** 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

NAME CLASS HOSTS ADDRESS PORTS AGE selenium-ingress <none> selenium.lt.dev.llamadev.net ab866e556790911ea8ab302807db7995-c9e51ea5083a9068.elb.us-east-1.amazonaws.com 80 42d

Open a web browser and type the grid URL http://pegasus.lt.dev.llamadev.net/ui or http://selenium.lt.dev.llamadev.net/ui Selenium-grid will be available in the browser and exposed to the internet	
Note the Chrome browser icon has a version number For example this is Chrome version 101 This is obtained from the docker image located in the **node-chrome-deployment.yaml** To update this image will be covered in the Selenium-grid read me	
To confirm that your new framework is healthy, type command clear Followed by command kubectl get all Confirm all pods are running and Ready column is populated with 1	Clear will refresh the screen See row below

```
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

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   <none>
4443/TCP,4442/TCP                           46m
service/selenium-srv         NodePort       172.20.227.92    <none>
4444:30001/TCP                              49m
```

```
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

NAME                                          DESIRED   CURRENT   READY   AGE
replicaset.apps/selenium-hub-79b667c579       1         1         1       52m
replicaset.apps/selenium-node-chrome-7fd9966cc6   10        10        10      42m
```

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
error: no context exists with the name:
"arn:aws:eks:us-east-1:927623560008:cluster/eks-lt-dev"
Type command
aws eks update-kubeconfig --name arn:aws:eks:us-east-1:927623560008:cluster

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

If you see
spec.ports[0].nodePort: Invalid value: 30001: provided port is already allocated error on kubectl create -f sgrid/hub-service.yaml command

Edit the yaml file, type command
vim sgrid/hub-service.yaml
Press i to enter edit mode, now modify nodePort value example: 30001 to 30002
Press esc to escape edit mode, now enter :wq to save file and retry the cloudshell command