Most Frequently faced Issues & Troubleshooting Steps

**Issues with UCP Environment**

1. If a service URL is not accessible in Docker UCP:

* Validate the configuration of the service. If it is fine, check ucp interlock proxy logs to find any possible errors.
* If no errors found in ucp interlock proxy logs or you can see ucp interlock proxy service not getting updated or the network of the application service is not present in ucp interlock proxy service then restart the service manually from dashboard as a temporary work around.
* After manually restarting the service, if containers are not getting restarted in ucp interlock proxy logs or they are getting killed constantly then scale down and scale up the service (scale up 2 containers for ITG/4 containers for Prod) using below mentioned commands in a UCP Manager node.

docker service scale ucp-interlock-proxy=0

docker service scale ucp-interlock-proxy=2/4

1. If you see some error like “Worker Connections are not enough” present in ucp interlock proxy logs:

* Try increasing the max connection configuration from 10k to 15k by running the below commands on Master Node. (In the 4th Step change MaxConnections from 10000 to 15000).

Note: Please do not increase the maximum connections to more than 20000

1. CURRENT\_CONFIG\_NAME=$(docker service inspect --format '{{ (index .Spec.TaskTemplate.ContainerSpec.Configs 0).ConfigName }}' ucp-interlock)
2. docker config inspect --format '{{ printf "%s" .Spec.Data }}' $CURRENT\_CONFIG\_NAME > config.toml
3. ll
4. vi config.toml

MaxConnections = 15000

1. NEW\_CONFIG\_NAME="com.docker.ucp.interlock.conf-$(( $(cut -d '-' -f 2 <<< "$CURRENT\_CONFIG\_NAME") + 1 ))"
2. docker config create $NEW\_CONFIG\_NAME config.toml
3. docker service update --config-rm $CURRENT\_CONFIG\_NAME --config-add source=$NEW\_CONFIG\_NAME,target=/config.toml ucp-interlock

**Issues with DCOS Environment**

1. If a user comes up with a request to restart their service in DCOS as they are not able to do so themselves.

* Please ask the user to stop and start their service. If they are unable to do so, you can try to stop and start their service with Mesos Dashboard first.
* If you are also not able to do so, then ask them to go to Marathon Dashboard <DCOS URL>/marathon and try to restart their service themselves.
* If they don't have access to the marathon dashboard then you can go to the dashboard and restart the service.

1. If a service is getting restarted continuously, maybe due to some Application Code issue and due to that user is not able to see logs of failed tasks.

* Please go to Marathon Dashboard and check that service.
* That service might be in Delayed State due to continuous failures/restarts.
* Click on the Reset Delay option.
* After that, the tasks which were earlier in the waiting state get started and you can see their logs in the Mesos Dashboard then.

**Issues with ELK**

1. Status Code - 401 Security Exception

* Whenever you see this 401 issue for elastic user or any other user, it means that `.security` index is red. You can check this when you run a https://elasticsearch.its.hpecorp.net:9200/\_cat/indices from the Jumphost.
* If you see a Master not discovered exception, try restarting the master node services and wait for the master nodes to rejoin

1. You can now login to Kibana dashboard as soon as the `.security` index is Yellow
2. Then monitor the shard allocation status using <https://elasticsearch.its.hpecorp.net:9200/_cat/allocation>

* If you see Authentication Exception for Elastic user, you cannot run any curl against any node from the elasticsearch cluster for further troubleshooting because the `.security` index which holds all the user and role maps is red.

In this case, you can make use of file realms that is enabled on master and coordinator nodes. Basically, file realms is a typical file-based authentication on a node that allows you to create a temporary user and saves it in a local file within that node.

1. SH into one of the master/coordinator nodes and create a temporary superuser, for example - $ ./bin/elasticsearch-users useradd test-user -p abc123 -r superuser
2. Verify that the user is created using - $ ./bin/elasticsearch-users list
3. Now elasticsearch cluster can be accessed using the test-user - $ curl -k –u test-user:abc123 <https://elasticsearch-itg.its.hpecorp.net:9200>
4. Now that we can access the cluster using test-user, check the cluster health and look for any shard allocation issues. E.g. You might notice some shards failed to allocate with the exception - "allocate\_explanation":"cannot allocate because allocation is not permitted to any of the nodes that hold an in-sync shard copy"
5. To retry all the failed shards again run - $ curl -XPOST –k –u test-user:abc123 ‘https://elasticsearch-itg.its.hpecorp.net:9200/\_cluster/reroute?retry\_failed’
6. This will fix the failed shards and the cluster will come back up green.
7. ELK Dashboard Down

If ELK is down with Kibana service or any other service failure due to continuous container restart then perform the following steps:

1. Login into all worker nodes and list the services using docker service ls | grep elk-stack
2. Scale down the failing service and clean up the exited containers of the service.
3. Scale up the service to bring it up and healthy - docker service scale servicename=1
4. User not able to see logs in ELK

Whenever user reports that they cannot see any events over a time range, please check if there is a calendar icon for that particular index. You should find it at the right corner on the Discover page. This icon will allow the user to navigate over a time range.

If it is missing, you can add this icon by deleting the current index pattern and create it again with the `@timestamp` field. To do this:

1. Go to Management > Index Patterns under Kibana Settings

2. Search for the current index pattern and delete it

3. Create the same index pattern again, this time with the `@timestamp` field as shown below

4. Once the index pattern is created, go to Discover Tab > Select the index pattern > Make sure there is a Calendar icon at the right corner for the index pattern