**Image update procedure**

1. Check the kanban bord to get the ticket information. Click on the ticket and expand it to see the issue’s details and description. Then assign the ticket it to you.
2. SSH into your synker box using either putty or mobaxterm. Then perform the following commands.

* Sudo su – maintuser “to assume the maintuser role.”
* Cd apps-synker/ “to get into that directory”
* ./app-synker-update\_v1.15.3.sh “ to execute the synker shell script”
* NOTE: before you run the sript make sure there is no “.tar.gz” files in the “apps-synker” directory. If there are, delete them first. Otherwise, they might throw off the synker script.

1. Identify and enter the number corresponding to the application you are updating.
2. Before pressing “enter to continue”, make sure the customer is CTF (certificate to field) approved. This means making sure the application has been approved to be deployed to a certain environment (test, staging, prod) and/or fabrics (nipr, sipr, jwics). To do this go to confluence and search for CTF, then select ODIN customer CtF Information (start it to save it for later for future reference). In the CtF documentation, search for the application you are updating and open its Ctf to read through it to make sure the right boxes are checked.
3. After you press enter, a file will open. Edit the file by putting “y” in front of the images you want to update and “n” in front of the images you do not want to update, additionally, update the image with the new image tag. Then save and close the file.
4. Select “y” to build the package. This step will build the new image
5. Select “y” to save it to AWS
6. Select”y” to choose the manual transfer option. Then select the number corresponding to the fabric where you want to push the newly built image.
7. Copy the text bellow the “text for MM Chat” and paste it into mattermost to notify the people in the SCIF that the new image is available.
8. Go to the AWS console and navigate to the S3 bucket service. Then go to the “sipr-mission-bootstrap” bucket. In that bucket find the “mission-apps” folder. Then find the app directory where the image was pushed, open it and select the image you pushed. Copy its S3 URI
9. SSH into the mission prod build box and assume the maintuser role as is step 2
10. cd into the “synker” directory and run this command to download the image from AWS S3 to this folder.

Aws s3 cp <URI\_of\_the\_image> . (the “.” at the end of the command indicates the current directory)

1. Now it is time to push the image to the registry. To do this, rum the command “ sudo podman load < <name\_of\_file\_downloaded>
2. Now start a container with the image pushed using this command “sudo podman run –network=’host’ synker:<app\_name> push -b=1”
3. Now go to keycloak and log in as an admin. To get the keycloak password, go to the aws console and navigate to the secret manager service and choose ODIN-MGMT-PROD, then retrieve the secret value.
4. In keycloak, first select the “baby-yoda” realm
5. Then go to groups, find the application you are working on, then go to members and add you as a member.
6. Now that you are a member, you can have access to the app to see if it is running by inputting the app URL into your browser. But to do that you need to get the app virtual service url. To do that you have two options.

* In keycloak, go to “clients”, search and select the app you are working on, scroll down to the “valid redirects URIs”, then copy the URI up to .mil
* Use this command “kubectl get virtualservices -A | grep <name\_of\_the\_app”

1. Now log into ArgoCd as an admin. To get the password, follow step 15, but this time go to ODIN-MISSION-PROD. Make sure your computer is whitelisted (in the AWS console, navigate to the “WAF & shield” service, then choose “IP sets”. Then add you IP address to “odin-admin” and “odin-non-admin”)
2. Now, you need to update the kustomization file. To do that, go back to your mission-prod build box and assume the git role using “sudo su – git”. Then navigste to the kustomization file folder. Cd sipr-mission-bootstrap/mission-manifests/<app\_name>/<name\_of the fabric/overlays/prod/kustomization.yaml. and update the file with the new image tag.
3. Go through the Git steps to push it to git.

* Git status ( you can use “git restore –staged <file” to unstage files ready to be commited
* Git add <file\_name>
* Git commit -m “message\_of\_your\_choice”
* Git push

1. Go to ArgoCD, sync “sipr-mission-prod”, then scroll down and search for the apps namespace, choose it and sync the “sipr-mission-prod-<name\_of\_app>-<name\_of\_app>
2. Access the app again to make sure it is running.
3. Now go back to the kanban board and drag the ticket step by step to the ‘need review” column.
4. Congratulations!!
5. Troubleshooting:

* **If your image doesn’t build**, try this to see if docker is full. Note that docker lives in /apps-synker/var/lib:

Sudo df -h or sudo du -h –max-depth=1 (df = disk free, du = disk usage, h = human readable, ‘**--max-depth=1**: This option limits the depth of the directory hierarchy for which disk usage is reported. In this case, **max-depth=1** means that **du** will only report the disk usage for the immediate directories in the current directory. These commands will show you the disk space usage in a human readable format). Execute the df command at the “/apps-synker/” level and the du command at the “/var/” level.

* Cd into the “docker” registry and do “sudo docker ps -a and also sudo docker images” to list all the containers and images
* Never delete the “registry:2” image that runs the “/entrypoint.sh /etc\_” command. It keeps track of all the synker sudo commands.
* Get rid of all the images with the <none> tag.
* for i in $(sudo docker ps -a | grep -I exited | egrep -v optimist | awk ‘{print $1}’); do sudo docker rm $i; done
  + to **rollback** to a previous image, do “git log” to get the list of previous commits. From that list grab the commit id, and do “git show <commit id>”

Grep -irn

Saga show dockerregistry |grep <name\_of\_app>

for i in $(ls -l | grep force- | awk '{print $9}' ); do aws s3 cp $i s3://dcw-odin-sc2s/ --acl bucket-owner-full-control; done