1. **Core**
   1. Create mongodb cluster.  
      Both stg and prod contexts (environments) already exist in mongodb.com.  
      Create cluster:
      1. Log into <https://cloud.mongodb.co> and navigate to stg context from pulldown.
      2. Click “Build a Cluster”
      3. Accept all defaults and enter “stg-cluster” for name.
      4. Click “Create cluster”
      5. When cluster is created, take note of all 3 shard names.
      6. White-list ip address of 0.0.0.0/0 in security for cluster.
   2. In local.js, module.exports.auth.casCallbackUrl.samlKeys.v0 should have the private key value for our SP. Before core became the saml SP for shibboleth, apache filled the SP role with its shibboleth module. Core now needs the private key that apache used to talk to shibboleth. This private key can be found on the related ec2 instance at: /etc/pki/tls/certs/sp-key.pem Replace the v0 value with the contents of sp-key.pem
   3. Upload core configurations to S3 (local.js. samlMeta.js, container env vars).
   4. Run Jenkins job to deploy core to target environment (no docker build, just docker run).
   5. The core container should modify fields in the institution and incommons collections in the mongo database. Make sure of this. IE: bu instition should have and issuer field added.
   6. Add “bu\_service\_iam” user to users collection in mongo with admin role
   7. Add “infopwc” user to users collection in mongo with admin role
   8. Add any other people to the users collection in mongo that should be admins
   9. Add “REST\_SERVICE\_USER” (username: “rest.svc.user”) to mongo users collection with “service” role.
   10. In browser visit <https://kuali-research.bu.edu> and see “Welcome to Kuali” with a “Sign-in” button.
   11. If all IDP/SP details are correctly configured, clicking “Sign-in” should take you to the shibboleth user/password entry page. After authentication, you should end up on the core landing page.
2. **Apache**
   1. Rerun apache Jenkins deployment job to pull in latest docker image from registry (has shibboleth plugin removed from baked in configuration).
   2. Execute the following line on one of the ec2 instances to verify the shibboleth plugin is no longer in use (plugin should be commented out).  
      docker exec apache-shibboleth cat /etc/httpd/conf.d/kc.conf
3. **KC**.
   1. Add “bu\_service\_iam” user to kim tables with super user role.
   2. Add the following params to kc-config.xml:  
        
      <param name="auth.base.url">${app.host}</param>  
      <param name="filter.restlogin.class">org.kuali.coeus.sys.framework.auth.AuthServiceFilter</param>  
      <param name="filtermapping.restlogin.1">/\*</param>  
      <param name="auth.core.enabled">true</param>  
      <param name="auth.system.token">PUT TOKEN HERE</param>
   3. Generate a key for “REST\_SERVICE\_USER” once the core website is up and paste its value into kc-config.xml at the “auth.system.token” param.
   4. Set the “auth.core.enabled” param in kc-config.xml to true
   5. Upload kc-config.xml with core configuration to s3
   6. Download kc-config.xml with update from s3 to each ec2-instance at /opt/kuali/main/config/kc-config.xml
   7. Restart both kuali-research containers.
   8. The users that were manually entered into the mongo database should be able to log in through shibboleth. All others will not.
4. Run the ETL job.