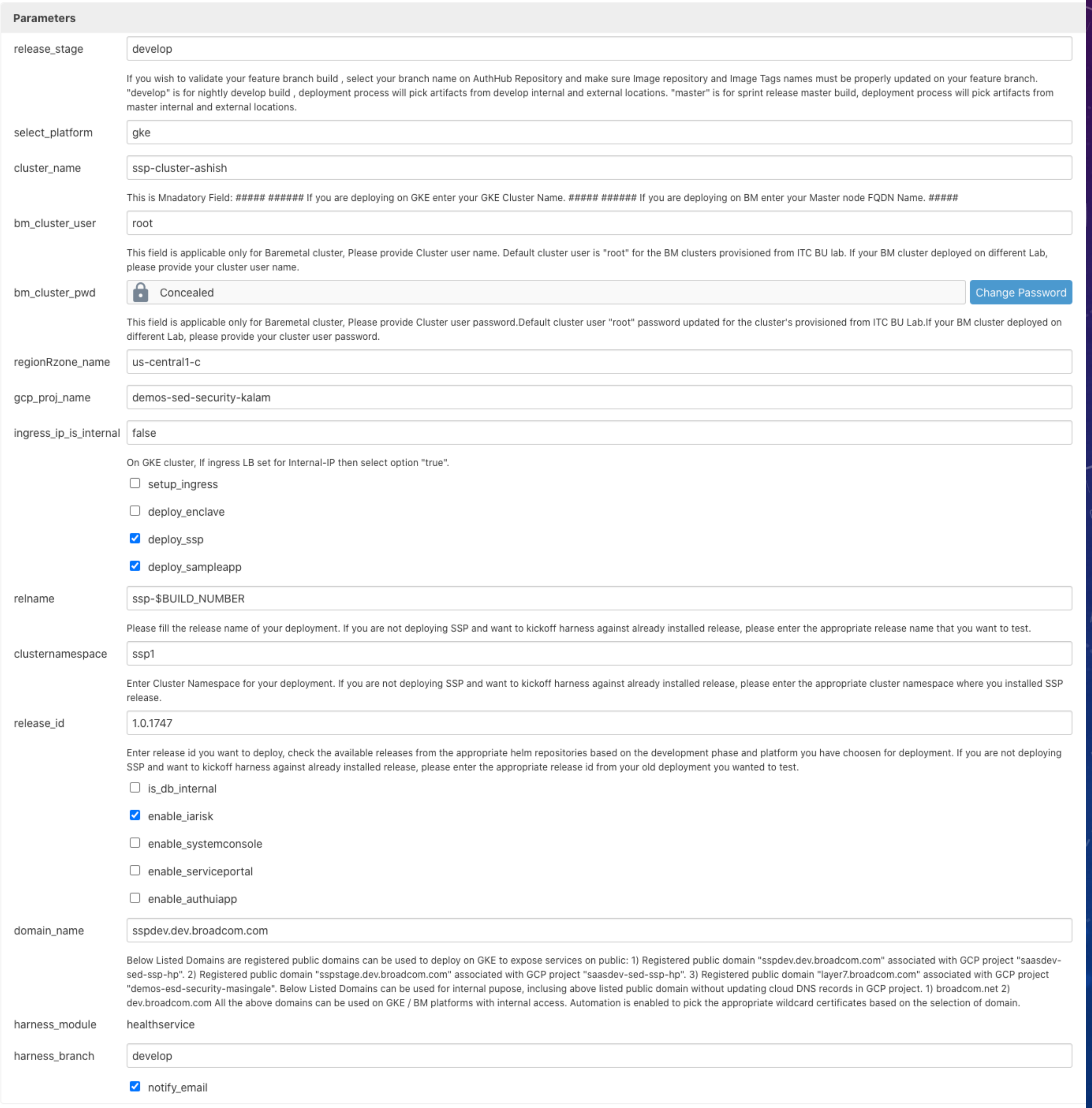
GKE Setup Instructions

By K.C.Ashish Kumar

Install via Jenkins Job

- http://authub-jenkins-srv.lvn.broadcom.net:8080/
- IAM-PUBLIC >> deploy-ngnx-enclave-ssp-on-gke-bm-cluster-kickoff-harness
- Build with Parameters (Refer Screenshot)
- The build normally takes 20-30 Minutes

Note:
For the first time deployment also select the below 2 options:
setup_ingress
deploy_enclave



Parameters

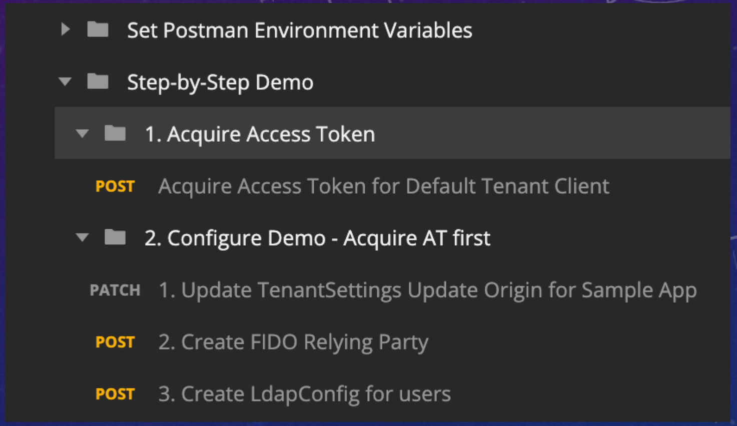
release_stage	develop
	If you wish to validate your feature branch build , select your branch name on AuthHub Repository and make sure Image repository and Image Tags names must be properly updated on your feature branch. "develop" is for nightly develop build , deployment process will pick artifacts from develop internal and external locations. "master" is for sprint release master build, deployment process will pick artifacts from master internal and external locations.
select_platform	gke
cluster_name	ssp-cluster-ashish
	This is Mnadatory Field: ##### ###### If you are deploying on GKE enter your GKE Cluster Name. ##### ###### If you are deploying on BM enter your Master node FQDN Name. #####
bm_cluster_user	root
	This field is applicable only for Baremetal cluster, Please provide Cluster user name. Default cluster user is "root" for the BM clusters provisioned from ITC BU lab. If your BM cluster deployed on different Lab, please provide your cluster user name.
bm_cluster_pwd	Concealed — Change Password
	This field is applicable only for Baremetal cluster, Please provide Cluster user password.Default cluster user "root" password updated for the cluster's provisioned from ITC BU Lab.If your BM cluster deployed on different Lab, please provide your cluster user password.
regionRzone_name	us-central1-c
gcp_proj_name	demos-sed-security-kalam
ingress_ip_is_internal	false
	On GKE cluster, If ingress LB set for Internal-IP then select option "true".
	☐ setup_ingress
	☐ deploy_enclave
	☑ deploy_ssp
	☑ deploy_sampleapp
relname	ssp-$BUILD_NUMBER
	Please fill the release name of your deployment. If you are not deploying SSP and want to kickoff harness against already installed release, please enter the appropriate release name that you want to test.
clusternamespace	ssp1
	Enter Cluster Namespace for your deployment. If you are not deploying SSP and want to kickoff harness against already installed release, please enter the appropriate cluster namespace where you installed SSP release.
release_id	1.0.1747
	Enter release id you want to deploy, check the available releases from the appropriate helm repositories based on the development phase and platform you have choosen for deployment. If you are not deploying SSP and want to kickoff harness against already installed release, please enter the appropriate release id from your old deployment you wanted to test.
	☐ is_db_internal
	☑ enable_iarisk
	☐ enable_systemconsole
	☐ enable_serviceportal
	☐ enable_authuiapp
domain_name	sspdev.dev.broadcom.com
	Below Listed Domains are registered public domains can be used to deploy on GKE to expose services on public: 1) Registered public domain "sspdev.dev.broadcom.com" associated with GCP project "saasdev-sed-ssp-hp". 2) Registered public domain "sspstage.dev.broadcom.com" associated with GCP project "saasdev-sed-ssp-hp". 3) Registered public domain "layer7.broadcom.com" associated with GCP project "demos-esd-security-masingale". Below Listed Domains can be used for internal pupose, inclusing above listed public domain without updating cloud DNS records in GCP project. 1) broadcom.net 2) dev.broadcom.com All the above domains can be used on GKE / BM platforms with internal access. Automation is enabled to pick the appropriate wildcard certificates based on the selection of domain.
harness_module	healthservice
harness_branch	develop
	☑ notify_email

Common COMMANDS:

- Install gcloud SDK - https://cloud.google.com/sdk/docs/
- Also install the “kubectl” component
- gcloud auth login
- gcloud config set project <PROJECT_ID>
- gcloud container clusters get-credentials <clusterName> --region us-central1-c --project <PROJECT_ID>
  command for connecting to the cluster
- kubectl get nodes -o wide
  command for getting the information about internal/external IP
- kubectl get pods,svc -n idstore
  command for getting information about ports for LDAP (here the namespace is “idstore”)
- kubectl get pods -n <namespace>
  command for getting all the pods in a specific namespace (Also can use pod argument instead of pods)
- kubectl describe pods -n ssp1
  command to get detailed information for pods in a specific namespace (Also can use pod argument instead of pods)
- kubectl describe pods <podName> -n <namespace>
  command to describe a specific pod within a specific namespace (Also can use pod argument instead of pods)
- kubectl --help
  General help
- kubectl get --help
  Help for the get argument
- kubectl edit deployment -n <NAMESPACE> <PODNAME_WITHOUT_REPLICA_ID>
  (For editing the deployment information like image etc.,.)
- kubectl scale deployment -n <NAMESPACE> <PODNAME_WITHOUT_REPLICA_ID> --replicas=XX
  XX=0 (For destroying all the POD instances)
  XX=1 (For creating a new POD instance)
- helm ls -A
  (For checking the various deployments)
- kubectl get ingress -A
  (For checking the various ingress hostnames)
- systemctl enable haproxy
  (For enabling haproxy)
- journalctl –u haproxy
  (For checking haproxy logs)
- reboot –reboot
  (For rebooting the machine)

Configure LDAP & Users

- Go to POSTMAN and execute the "Step by Step Demo >> Configure Demo" i.e. execute the steps in screenshot.
  This will create the LDAP Config for the GKE Environment.
  Note: It is mandatory to execute all the shown steps i.e.
  1.1, 2.1, 2.2, 2.3



Set Postman Environment Variables

Step-by-Step Demo

1. Acquire Access Token

POST Acquire Access Token for Default Tenant Client

2. Configure Demo - Acquire AT first

PATCH 1. Update TenantSettings Update Origin for Sample App

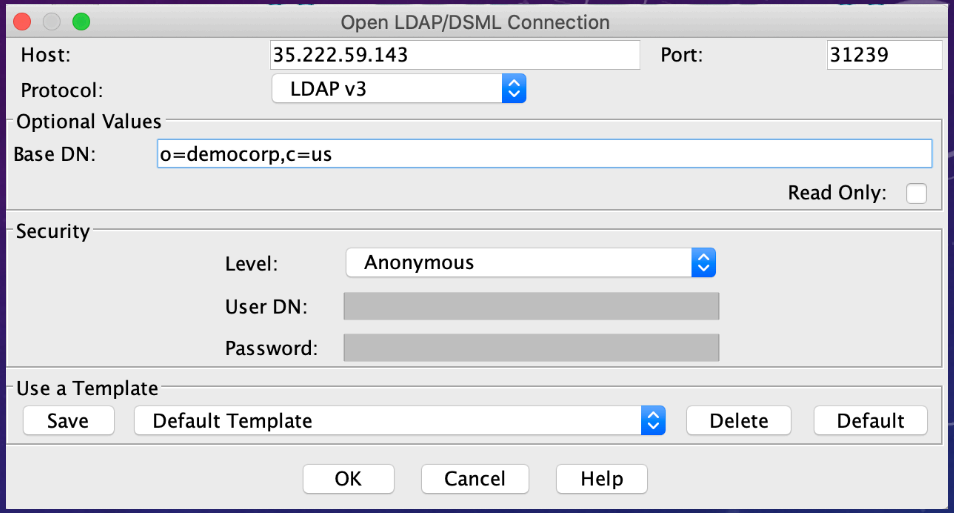
POST 2. Create FIDO Relying Party

POST 3. Create LdapConfig for users

Configure LDAP & Users

- Install Java & JXplorer, Run JXplorer
- Use the external IP & the port obtained from the kubectl command
  ExternalIP:
  `kubectl get nodes -o wide`
  LDAP Port:
  `kubectl get pods,svc -n idstore`
- Refer the 1st screenshot for Base DN
- Save the settings as a template e.g. Default Template
- Connect (Press OK) (Refer 2nd screenshot for screen after successful connection)

Contd . . .



Host: 35.222.59.143   Port: 31239

Protocol: LDAP v3

Optional Values

Base DN: o=democorp,c=us

Read Only: □

Security

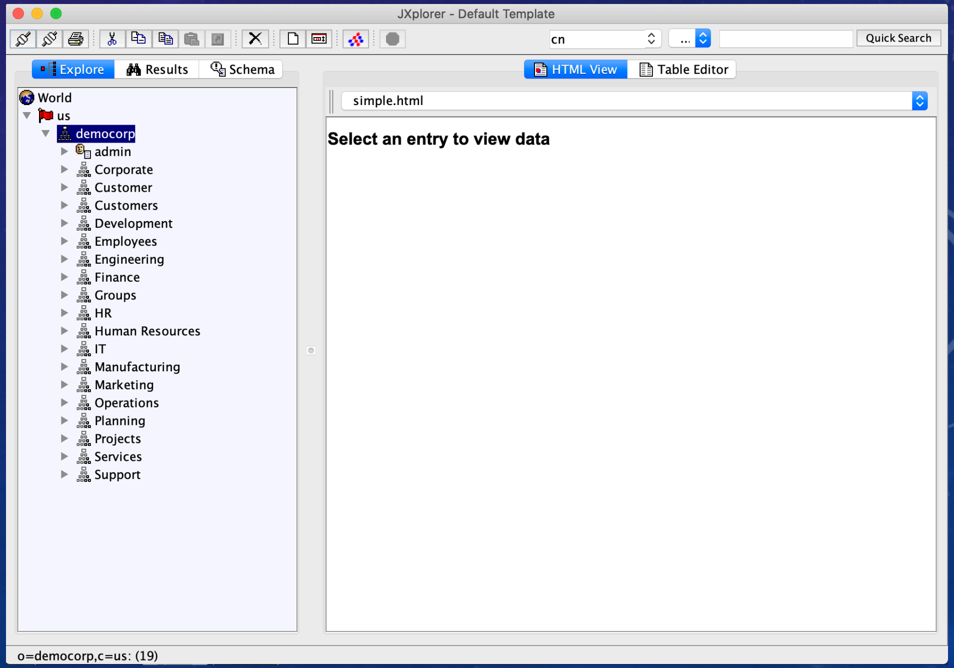
Level: Anonymous

User DN:

Password:

Use a Template

Default Template



- World
  - us
    - democorp
      - admin
      - Corporate
      - Customer
      - Customers
      - Development
      - Employees
      - Engineering
      - Finance
      - Groups
      - HR
      - Human Resources
      - IT
      - Manufacturing
      - Marketing
      - Operations
      - Planning
      - Projects
      - Services
      - Support

simple.html

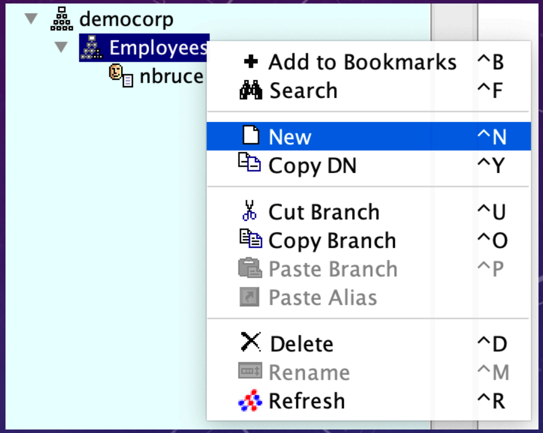
Select an entry to view data

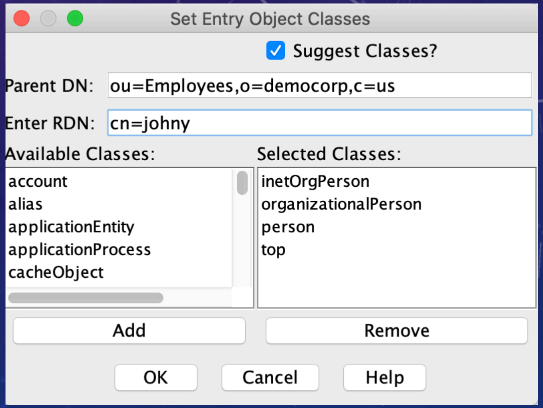
o=democorp,c=us: (19)

Configure LDAP & Users

- In the search bar, search for an existing user e.g. nbruce
- In the results, right-click on Employees & select 'New'
- Refer the 2nd screenshot for values, replace 'johny' with the desired username. Click OK
- In the next screen, fill out the mandatory values like 'sn' (Surname), userPassword.
- For userPassword, select type as SHA-256
- Click "Submit"
- Go to HTML View and fill out other fields if needed like phone, email & click Submit
- Sometimes the *password* isn't set properly, so use the HTML View to set the password again and hit Submit.

Contd . . .




Set Entry Object Classes

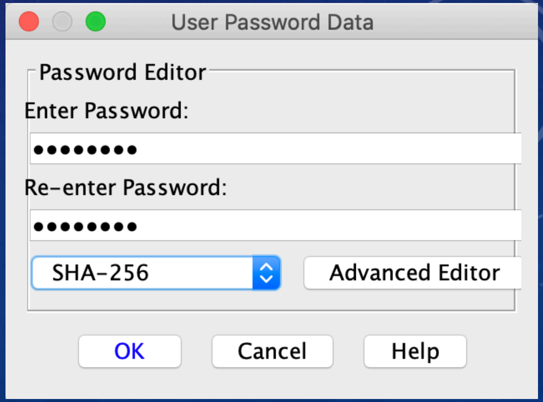
☑ Suggest Classes?

Parent DN: ou=Employees,o=democorp,c=us

Enter RDN: cn=johny

Available Classes:
account
alias
applicationEntity
applicationProcess
cacheObject

Selected Classes:
inetOrgPerson
organizationalPerson
person
top


User Password Data

Password Editor

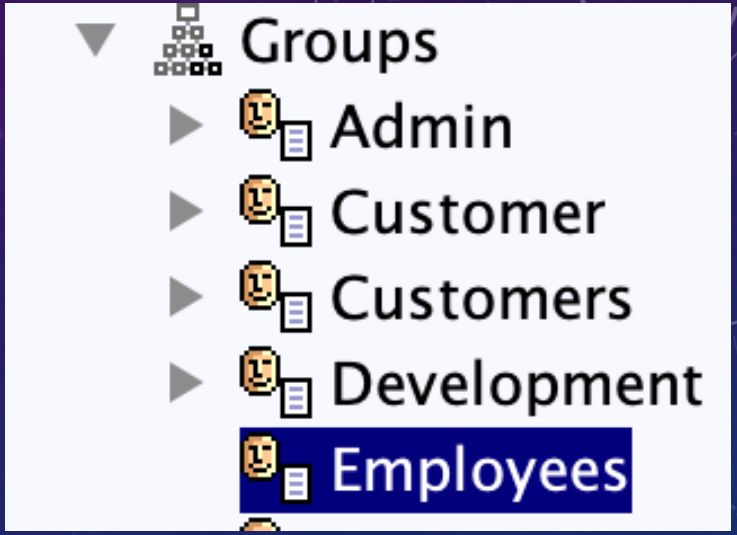
Enter Password: ••••••••

Re-enter Password: ••••••••

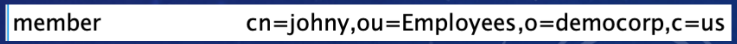
SHA-256

Configure LDAP & Users

- Go to the “Explore” tab and select “Groups >> Employees” (Refer: screenshot)



- In the right “Table Editor” view, add a new “member” entry for the new user “johny” like the existing entries. (Refer: 2nd screenshot)



member cn=johny,ou=Employees,o=democorp,c=us

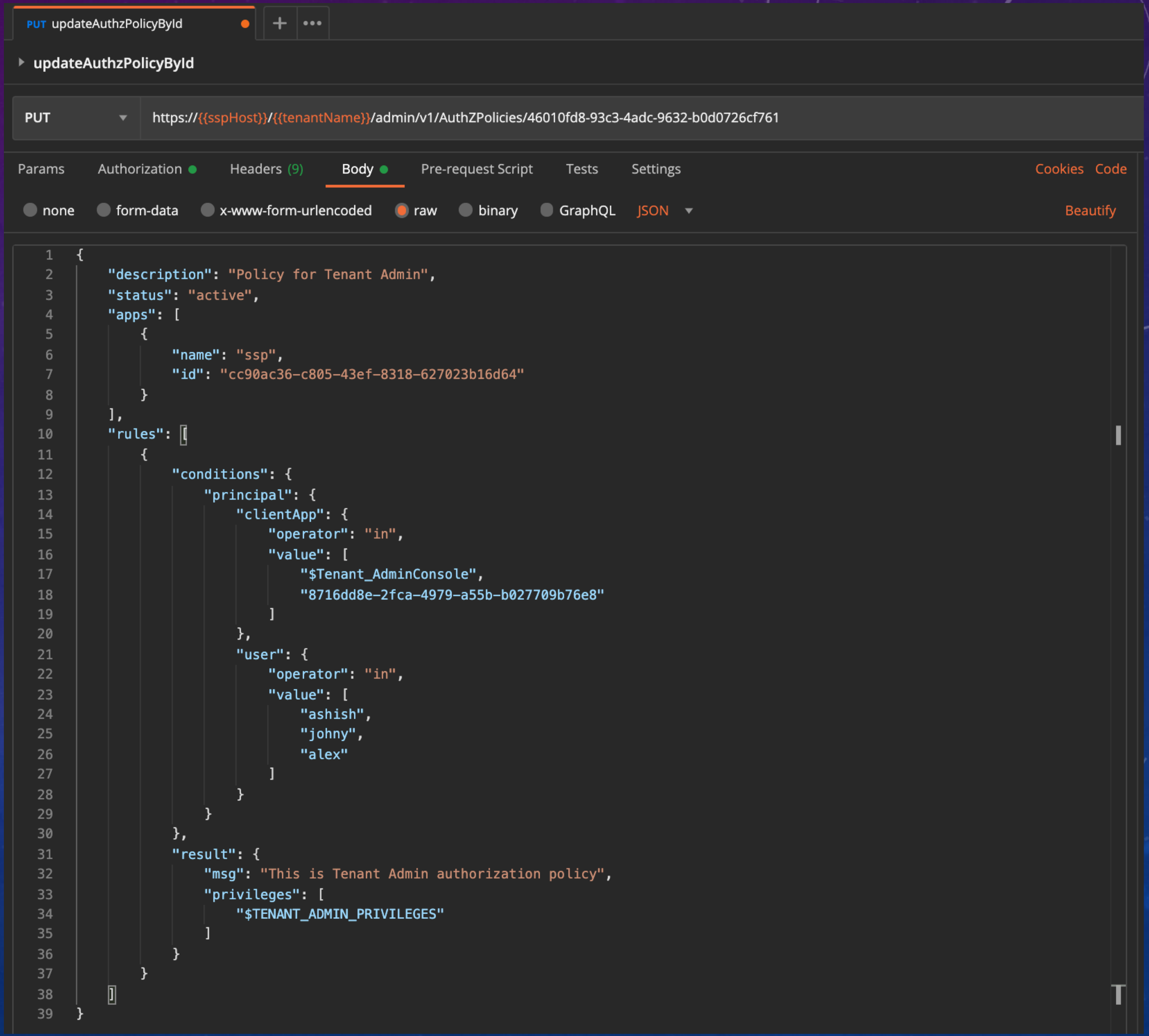
- Click Submit

- The new user account “johny” is now ready.

Adding Users to Authz Policy

- Refer the example screenshot to add “user” inside the “principal” line# 21 – 28

  POSTMAN Script: updateAuthzPolicyById



PUT https://{{sspHost}}/{{tenantName}}/admin/v1/AuthZPolicies/46010fd8-93c3-4adc-9632-b0d0726cf761

```
 1  {
 2      "description": "Policy for Tenant Admin",
 3      "status": "active",
 4      "apps": [
 5          {
 6              "name": "ssp",
 7              "id": "cc90ac36-c805-43ef-8318-627023b16d64"
 8          }
 9      ],
10      "rules": [
11          {
12              "conditions": {
13                  "principal": {
14                      "clientApp": {
15                          "operator": "in",
16                          "value": [
17                              "$Tenant_AdminConsole",
18                              "8716dd8e-2fca-4979-a55b-b027709b76e8"
19                          ]
20                      },
21                      "user": {
22                          "operator": "in",
23                          "value": [
24                              "ashish",
25                              "johny",
26                              "alex"
27                          ]
28                      }
29                  }
30              },
31              "result": {
32                  "msg": "This is Tenant Admin authorization policy",
33                  "privileges": [
34                      "$TENANT_ADMIN_PRIVILEGES"
35                  ]
36              }
37          }
38      ]
39  }
```

Setup Index-Patterns in Kibana

- Open the “kibana” url in the browser.
- Click on “Discover” or the “D” in top bar and click “Manage Spaces”
- Select option “Index Patterns” and then select the appropriate index i.e. ssp_log* (or) ssp_audit* and click Next
- Create a filter with @timestamp and save.