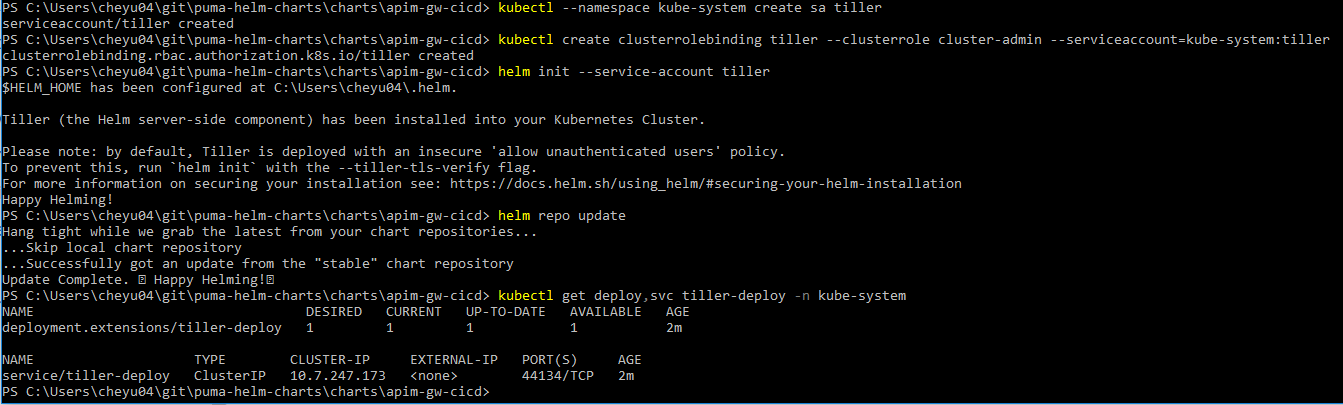
## Prerequisites

1. GitHub
   1. A public GitHub account ([github.com](https://github.com/), not the Broadcom GitHub site)
   2. Git is installed on your machine
2. These values will be required during installation. Generate your own for the blanks (lower, upper, numbers, symbols). The placeholders for these values will be highlighted in the instructions.
   1. Sandbox name (from instructions, e.g. sandbox2)
   2. Cluster subdomain (from instructions, e.g. sandbox2.apimgcp.com)
   3. Cluster region (from instructions, e.g . us-west1)
   4. Jenkins
      1. Username: admin
      2. Password: \_\_\_\_\_\_\_\_\_\_\_
   5. Nexus
      1. Username: admin
      2. Password: \_\_\_\_\_\_\_\_\_\_\_
   6. Grafana
      1. Username: admin
      2. Password: \_\_\_\_\_\_\_\_\_\_\_
   7. GW base Docker image repository:
      1. Username: admin
      2. Password: b2ycT$^KJpg10Z%wXw7S
   8. Gateway license file path (license sent in instruction email)
   9. Jenkins service account name
      1. apim-kubernetes-jenkins@api-management-178215.iam.gserviceaccount.com
   10. CICD Chart Helm release name
       1. apim-gw-cicd

## Tools and Environment Setup

1. Download and install the gcloud CLI
   1. <https://cloud.google.com/sdk/?&_ga=2.49174565.-2032780738.1541611901#download>
      1. Login with the sandbox account
      2. When prompted for a project, select the existing project named
2. Install kubectl (Kubernetes command-line tool)
   1. gcloud components install kubectl
3. Download and install Helm
   1. Windows, Mac, and Linux instructions:
      1. <https://github.com/helm/helm/blob/master/docs/install.md>
   2. (Alternative, if the above is not possible) Windows manual install method:
      1. <https://medium.com/@JockDaRock/take-the-helm-with-kubernetes-on-windows-c2cd4373104b>
4. Get credentials and connect to the Kubernetes cluster
   1. gcloud beta container clusters get-credentials apim-kubernetes-<sandbox name> --region <cluster region> --project api-management-178215
5. Install Tiller on Kubernetes cluster
   1. Add a service account in the kube-system namespace for Tiller to use
      1. kubectl --namespace kube-system create sa tiller
   2. Create a cluster role binding for the service account to grant Tiller cluster admin privileges
      1. kubectl create clusterrolebinding tiller --clusterrole cluster-admin --serviceaccount=kube-system:tiller
   3. Initialize helm using the tiller service account
      1. helm init --service-account tiller
   4. Update the local repo for your Helm installation
      1. helm repo update
   5. Verify that Tiller is installed correctly on the cluster
      1. kubectl get deploy,svc tiller-deploy -n kube-system



1. Fork the example design time repository
   1. Go to the [Design Time Repository](https://github.com/kmienata/puma-ephemeral-gateway-config)
   2. Click Fork on the right top corner
   3. Clone your forked version of the repository
      1. git clone [https://github.com/<repository name>/puma-ephemeral-gateway-config.git](https://github.com/)
2. Update Jenkinsfile configurations
   1. Open Jenkinsfile in the cloned repository
   2. Change the GIT\_REPOSITORY to your forked repository link
      1. GIT\_REPOSITORY = '<https://github.com/><repository name>/puma-ephemeral-gateway-config'
   3. Change NEW\_IMAGE\_REGISTRY\_HOSTNAME to the nexus repository on your sandbox environment (but leave BASE\_IMAGE\_REGISTRY\_HOSTNAME unchanged)
      1. NEW\_IMAGE\_REGISTRY\_HOSTNAME = ‘docker.<sandbox name>.apimgcp.com’
   4. Commit your changes to your fork on public github
3. Clone the POC Helm Charts repository
   1. git clone <https://github.gwd.broadcom.net/ESD/puma-helm-charts.git>
4. Change your working directory to “puma-helm-charts/charts/”

## Jenkins & Nexus

To deploy the cert-manager, Jenkins & Nexus on the Kubernetes cluster, follow the steps below:

1. Ensure your working directory is “puma-helm-charts/charts/”
2. Create a key for the pre-defined DNS Administrator serviceaccount
   1. gcloud iam service-accounts keys create ./clouddns.key.json --iam-account apim-kubernetes-sandbox-dns@api-management-178215.iam.gserviceaccount.com --project api-management-178215
3. Install the Chart for cert-manager
   1. helm dep build apim-gw-cert-manager
   2. helm install apim-gw-cert-manager --name=apim-gw-cert-manager --set acmeEmail=apim<sandbox name>@gmail.com --set-file clouddnsServiceAccountKeyFile=clouddns.key.json --set prodCertificateName=apim-prod --set prodCertificateSecretName=apim-tls --set certDomainName="\*.<cluster subdomain>" --set prodEnabled=true --set stagingEnabled=false
4. Install the Chart for Jenkins and Nexus
   1. helm dep build apim-gw-cicd
   2. helm install apim-gw-cicd --name apim-gw-cicd --set sonatype-nexus.nexusProxy.env.nexusDockerHost=docker.<cluster subdomain> --set sonatype-nexus.nexusProxy.env.nexusHttpHost=nexus.<cluster subdomain> --set sonatype-nexus.ingress.tls.secretName=apim-tls --set sonatype-nexus.ingress.annotations."kubernetes\.io\/ingress\.global-static-ip-name"=apim-<sandbox name>-nexus-ip --set jenkins.Master.AdminPassword="<jenkins password>" --set nexusAdminPassword=<nexus password> --set jenkins.Master.HostName=jenkins.<cluster subdomain> --set jenkins.Master.Ingress.TLS[0].hosts[0]=jenkins.<cluster subdomain> --set jenkins.Master.Ingress.TLS[0].secretName=apim-tls --set jenkins.Master.Ingress.Annotations."kubernetes\.io\/ingress\.global-static-ip-name"=apim-<sandbox name>-jenkins-ip

## InfluxDB + Grafana

To deploy InfluxDB & Grafana on the Kubernetes cluster, follow the steps below:

1. Ensure your working directory is “puma-helm-charts/charts/”
2. Install the Chart for InfluxDB and Grafana.
   1. helm dep build apim-service-metrics-runtime
   2. helm install apim-service-metrics-runtime --name=apim-service-metrics-runtime --set grafana.adminPassword="<grafana password>" --set grafana.ingress.hosts[0]="grafana.<cluster subdomain>" --set grafana.ingress.tls[0].hosts[0]="grafana.<cluster subdomain>" --set grafana.ingress.tls[0].secretName="apim-tls" --set grafana.ingress.annotations."kubernetes\.io\/ingress\.global-static-ip-name"=apim-<sandbox name>-grafana-ip

## Log Stack

To deploy Filebeat, Elasticsearch & Kibana on the Kubernetes cluster, follow the steps below:

1. Ensure your working directory is “puma-helm-charts/charts/”
2. Install the Chart for Elasticsearch and Kibana
   1. helm dep build apim-logs-runtime
   2. helm install apim-logs-runtime --name=apim-logs-runtime
3. Install the Chart for Filebeat, which will collect and forward logs to Elasticsearch
   1. helm dep build apim-logs-shipper
   2. helm install apim-logs-shipper --name=apim-logs-shipper

## Jenkins Manual Steps

1. **For Linux and Mac Users ONLY**:
   1. Ensure your working directory is “puma-helm-charts/scripts/” and run the following script
      1. ./jenkins-setup.sh -s <jenkins service account name> -c <cicd release name> -h <gateway license file path>

Note : arguments for the above command

-s → Mandatory (Service account name)

-c → Mandatory (CICD release name)

-h → Mandatory (Gateway license file path)

-f → Optional (Google-cloud-sdk tar name)

1. **For Windows Users ONLY**:
   1. Find the name of the Jenkins pod in your cluster (starts with “apim-gw-cicd-jenkins-”)
      1. kubectl get pods | grep jenkins
   2. Copy your license inside the Jenkins container.
      1. kubectl cp "<gateway license path>" "<jenkins pod name>":/var/jenkins\_home/license.xml
   3. Connect to the Jenkins container.
      1. kubectl exec -it <jenkins pod name> -- /bin/bash
   4. Install GCloud SDK + Kubectl extension on the Jenkins container so Jenkins can update the Gateway deployment with the new image.
      1. curl -O <https://dl.google.com/dl/cloudsdk/channels/rapid/downloads/google-cloud-sdk-226.0.0-linux-x86_64.tar.gz>
      2. tar zxvf google-cloud-sdk-226.0.0-linux-x86\_64.tar.gz google-cloud-sdk
      3. ./google-cloud-sdk/install.sh --usage-reporting false --path-update true --command-completion true --rc-path /root/.bashrc --additional-components kubectl beta
      4. exec bash
   5. Set up GCloud credentials to use kubectl.
      1. gcloud auth login (follow the instructions on command line)
      2. When you run this command you will get this message

- Do you want to continue (Y/n)? - Enter Y.

* + 1. Copy url paste it in browser. Select sandbox account and copy code and paste it back to terminal
    2. gcloud beta container clusters get-credentials apim-kubernetes-<sandbox name> --region <cluster region> --project api-management-178215
    3. mv ./google-cloud-sdk/bin/kubectl /usr/local/bin/kubectl
    4. Verify install:
       1. kubectl get pods
    5. Logout:
       1. exit

1. Set up Jenkins configurations on Jenkins UI (pass in credentials through Jenkins parameters) and webhook for github
   1. Get Jenkins UI URL:
      1. Go to <https://console.cloud.google.com>
      2. Navigate: Kubernetes Engine -> Services
      3. Endpoint for:
         1. Name: apim-gw-cicd-jenkins
         2. Type: Load balancer
         3. Cluster: apim-kubernetes-<sandbox name>
   2. Configure Webhook for Github to link to Jenkins
      1. Go to your Github *puma-ephemeral-gateway-config* repository
      2. Go to **Settings** tab
      3. Go to **Webhooks** section, “Add Webhook”
      4. In the Payload URL, enter http://<jenkins endpoint>/github-webhook/ (e.g <http://35.236.36.147:8080/github-webhook/>)
      5. Leave everything else as it is and click “Add Webhook”
   3. Login to Jenkins
      1. Follow step a
      2. Click the endpoint
   4. Click on Jenkins Build Job (e.g. test-job)
   5. Click on **Configure** on the left hand side
   6. Change GitHub Project URL to your forked repository
      1. Project url: https://github.com/<repository name>/puma-ephemeral-gateway-config.git/
   7. Select **This Project is Parameterized** under **General** tab
   8. Set the credentials for the registry where github project will pull the image from and push the image to e.g. docker registry. Add four new **String** parameters:
      1. This is the username and password for the registry which holds the base gateway layer image.
         1. **Name**: BASE\_IMAGE\_REGISTRY\_USER

**Default Value**: < GW image username>

* + - 1. **Name**: BASE\_IMAGE\_REGISTRY\_PASSWORD

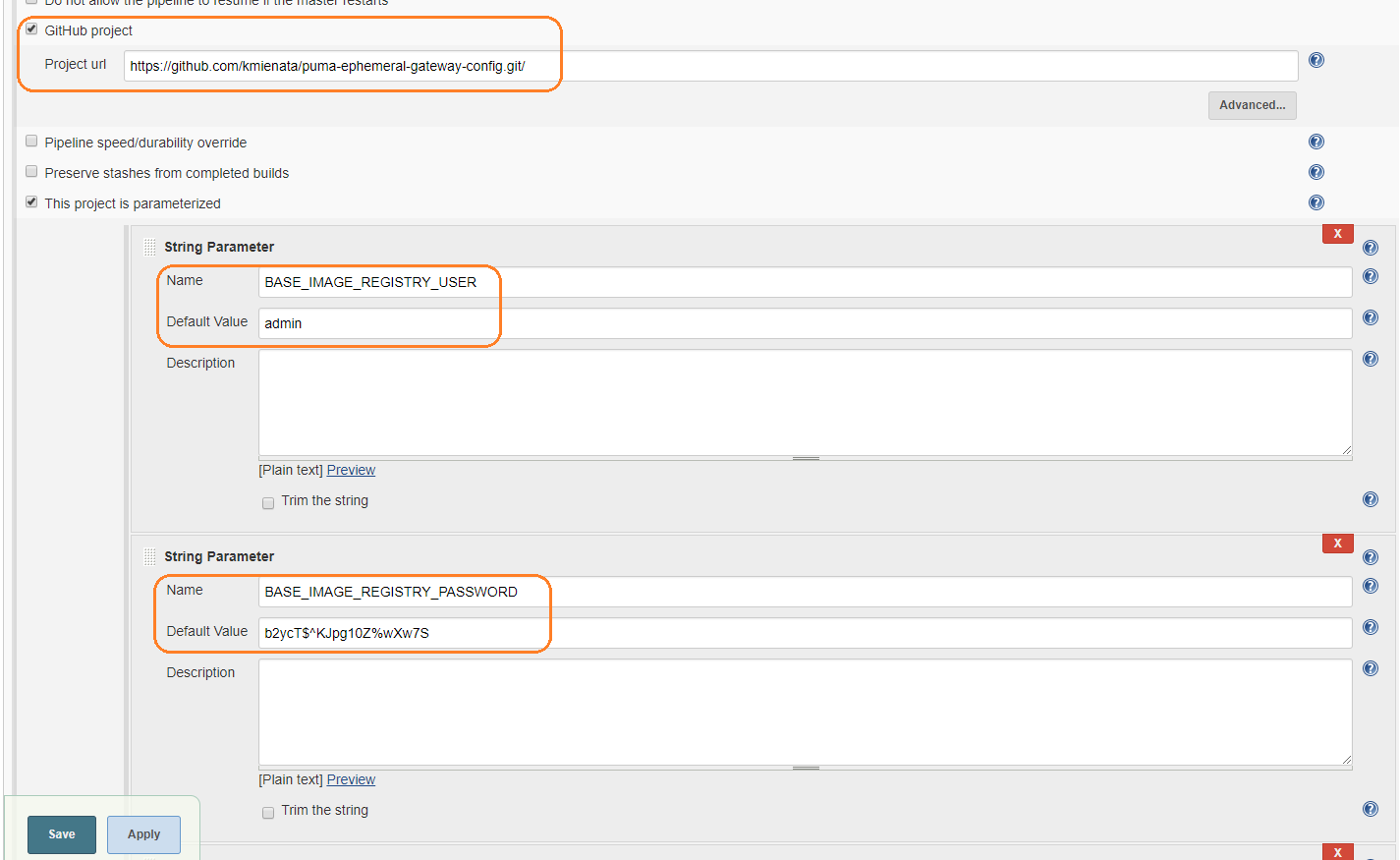
**Default Value**: < GW image password>

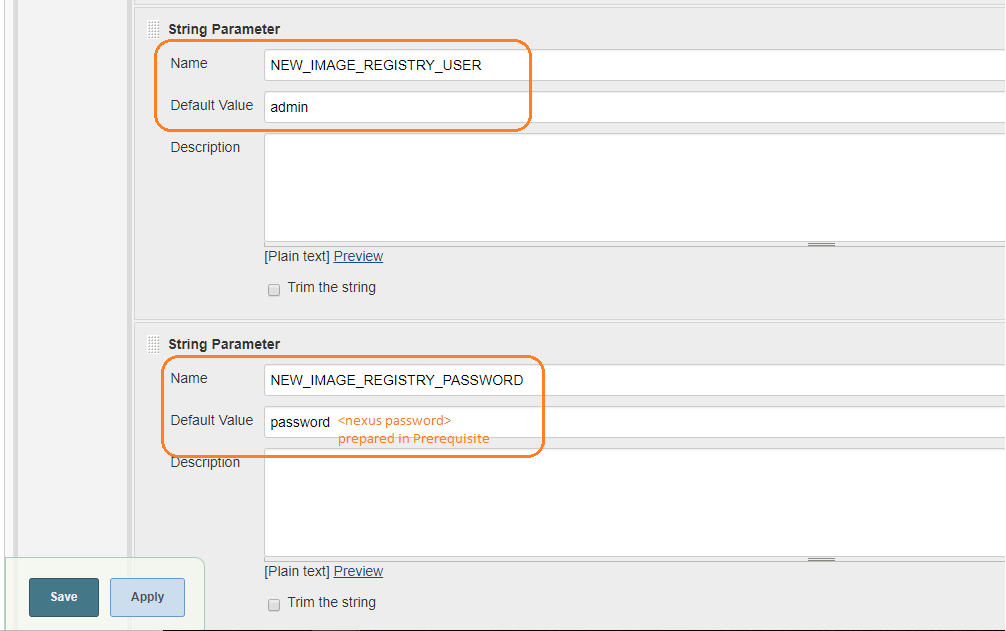
* + 1. This is the username and password for the nexus registry in your cluster which holds the new gateway layer image.
       1. **Name**: NEW\_IMAGE\_REGISTRY\_USER

**Default Value**: <Nexus username>

* + - 1. **Name**: NEW\_IMAGE\_REGISTRY\_PASSWORD

**Default Value**: <Nexus password> prepared in *Prerequisites*





* 1. Change the repository link where Jenkinsfile is located
     1. Go to the **Pipeline** section
     2. Change Repository URL to your forked github repository
        1. Repository URL: <https://github.com/><repository name>/puma-ephemeral-gateway-config.git
  2. Save the changes

1. Run the job build by clicking **Build With Parameters**, or going to this URL: <http://jenkins>.<cluster subdomain>:8080/job/test-job/build
   1. Expected failure on step “Update Gateway”

## Gateway + Hazelcast

To deploy the Gateway & Hazelcast on the Kubernetes cluster, follow the steps below:

1. Ensure your working directory is “puma-helm-charts/charts/”
2. Install the gateway Chart:
   1. helm dep build gateway
   2. helm install gateway --name "apim-gw" --namespace "default" --set-file "gateway.license=<gateway license file path>" --set imageCredentials.name="docker.<cluster subdomain>" --set imageCredentials.registry="docker.<cluster subdomain>" --set imageCredentials.password="<nexus password>" --set image.imageName="docker.<cluster subdomain>/repository/docker-hosted/gateway" --set image.tag="v1" --set influxdb.host="apim-service-metrics-runtime-influxdb" --set hazelcast.enabled=true

### You have now completed installing the POC solution!

## Quick guide to the environment

These are extra steps you can take to play around with the Gateway and Design Time:

1. Using the gateway:
   1. Endpoints:
      1. curl -k https://gateway.<cluster subdomain>:443/quota
         1. Service with quota of 10 hits per minute
      2. curl -k https://gateway.<cluster subdomain>:443/echo
         1. Echos “HI!”
         2. Generates audits and logs
2. Viewing logs:
   1. Get Kibana’s IP:
      1. Go to <https://console.cloud.google.com>
      2. Navigate: Kubernetes Engine -> Services
      3. Endpoint for:
         1. Name: apim-logs-runtime-kibana
         2. Type: Load balancer
         3. Cluster: apim-kubernetes-<cluster subdomain>
   2. Setting up kibana, navigate to Kibana:
      1. Management -> Index Patterns
         1. Index Pattern: \*
         2. Time Filter field name: @timestamp
   3. Viewing logs:
      1. Discover
      2. Add filter: log.message exists
      3. Add filter: kubernetes.labels.app is gw-default
3. Viewing metrics:
   1. Go to https://grafana.<cluster subdomain>
   2. Browse:
      1. Home->Gateway Service Metrics
4. Using the Design Time Repository to make Gateway Configurations (more information about design time: <https://github.com/ca-api-gateway/gateway-developer-plugin>)
   1. Ensure you have a development environment gateway up and running
   2. Make the Gateway configurations through policy manager on your dev environment
   3. Ensure your working directory is “puma-ephemeral-gateway-config/”
   4. Ensure the *GatewayConnection* url in build.gradle file is pointed to your dev gateway environment
      1. url = 'https://<gateway ip>:8443/restman'
   5. Run the gradle export plugin
      1. ./gradlew clean-export
      2. ./gradlew export
   6. Change NEW\_IMAGE\_TAG in Jenkinsfile every time you make configurations (commits to the design time repository) to the Gateway
      1. NEW\_IMAGE\_TAG = ‘<image version>’
   7. Commit and push your changes to your forked repository