

PCEI R5 API Document

- [PCEI Release 5 APIs](#)
 - [Kubernetes Edge Cluster Registration API](#)
 - [Terraform Plan Execution API](#)
 - [Helm Chart Onboarding for Service and App Registration](#)
 - [Service Instance creation and App deployment onto target Kubernetes clusters API](#)
- [PCEI R5 API Postman Collection](#)

PCEI R4 API Document is located at this [link](#).

PCEI Release 5 APIs

North Bound Interface APIs for:

1. **Kubernetes Edge Cluster Registration** into Multi-Cluster Orchestrator (ONAP/EMCO). This API will allow to retrieve the K8S cluster config from Git and onboard it onto the orchestrator's cluster provider registry.
2. **Terraform Plan Execution** against target providers (Cloud, Equinix, Openstack). This API will allow to retrieve the Terraform plans from a Git repository and run them from the orchestrator's workflow engine (CDS).
3. **Helm Chart Onboarding into ONAP for Service and App Registration**. This API will allow to retrieve Helm charts from a Git repository and create a Service and Application definitions within the Orchestrator for a specific tenant.
4. **Service Instance creation and App deployment onto target Kubernetes clusters**. This API will allow to create Service Instances based on the Services/Apps defined (as in point 3 above) in the orchestrator and deploy the Apps onto the target Kubernetes clusters (onboarded using point 1 above).

Kubernetes Edge Cluster Registration API

```

# You can refer to the below EMCO REST API link to get more details about the JSON payload
# https://wiki.onap.org/display/DW/V2+API+Specification
# Note:- You should update the attribute values for USER_INPUT_XXX parameters
# USER_INPUT_EMCO_V2_API_URL_PREFIX : here update the amcop installed machine ip with portno(30480).
# ex : http://<amcop server ip>:30480/v2
# USER_INPUT_GIT_URL - git project url where kubconfig file exists
# ex : https://gitlab.com/api/v4/projects/27142906/repository/files/edge-k8s-pcei-azure-config/raw?
ref=development
# USER_INPUT_GIT_DOWNLOAD_FOLDER : This shouldn't be modified. Always use "/opt/app/onap/blueprints/deploy
/kubeconfig".
# USER_INPUT_GIT_ACCESS_TOKEN - refer main readme file for "how to generate git access token" steps.

$ vi request-payload.json
{
  "actionIdentifiers": {
    "mode": "sync",
    "blueprintName": "emco-api-executor",
    "blueprintVersion": "1.0.0",
    "actionName": "emco-k8s-cluster-register-workflow"
  },
  "payload": {
    "emco-k8s-cluster-register-workflow-request": {
      "emco-k8s-cluster-register-workflow-properties": {
        "emco-v2-api-url-prefix": "USER_INPUT_EMCO_V2_API_URL_PREFIX",
        "git-url": "USER_INPUT_GIT_URL",
        "git-access-token": "USER_INPUT_GIT_ACCESS_TOKEN",
        "git-download-folder": "/opt/app/onap/blueprints/deploy/kubeconfig",
        "emco-v2-cluster-provider-parameters": {
          "cluster-providers-payload": {
            "metadata": {
              "name": "USER_INPUT_NAME",
              "description": "USER_INPUT_DESCRIPTION",
              "userData1": "USER_INPUT_USERDATA1",
              "userData2": "USER_INPUT_USERDATA2"
            }
          }
        },
        "emco-v2-cluster-parameters": {
          "cluster-providers-cluster-payload": {
            "metadata": {
              "name": "USER_INPUT_NAME",
              "description": "USER_INPUT_DESCRIPTION",
              "userData1": "USER_INPUT_USERDATA1",
              "userData2": "USER_INPUT_USERDATA2"
            }
          }
        },
        "emco-v2-cluster-label-parameters": {
          "cluster-label-payload": {
            "label-name": "USER_INPUT_LABEL_NAME"
          }
        }
      }
    },
    "commonHeader": {
      "subRequestId": "143748f9-3cd5-4910-81c9-a4601ff2ea58",
      "requestId": "e5eb1f1e-3386-435d-b290-d49d8af8db4c",
      "originatorId": "SDNC_DG"
    }
  }
}

```

Terraform Plan Execution API

```

# Note that "terraform-action" attribute value should be "apply"
# USER_INPUT_CDS_PY_EXEC_POD_IP : Use below command to get the CDS pyexecutor IP

```

```
# kubectl get pods -n onap -o wide | grep dev-cds-py-executor | awk '{print $6}'
# USER_INPUT_GIT_URL ex . "https://gitlab.com/api/v4"
# USER_INPUT_GIT_ACCESS_TOKEN - refer main readme to generate the git access token.
# USER_INPUT_GIT_PROJECT_ID : Git terraform project id. ex. 26901776
# USER_INPUT_GIT_BRANCH - git branch name ex. "development"
# USER_INPUT_GIT_DOWNLOAD_FOLDER - Always give "/opt/app/onap/blueprints/deploy"
# USER_INPUT_GIT_ARCHIVE_FILE_NAME - any tar file name ex. "terraform-plans-poc.tar.gz"
# USER_INPUT_TERRAFORM_VARS_FILE_NAME - override file from git project ex. "terraform.tfvars"
# USER_INPUT_TERRAFORM_PLAN_FOLDER - give correct git terraform plan folder name ex ."demo-azure-connectivity"
```

```
{
  "commonHeader": {
    "originatorId": "System",
    "requestId": "123456",
    "subRequestId": "1234-12234"
  },
  "actionIdentifiers": {
    "blueprintName": "terraform-plan-executor",
    "blueprintVersion": "1.0.0",
    "actionName": "remote-python",
    "mode": "sync"
  },
  "payload": {
    "remote-python-request": {
      "cds-py-exec-pod-ip": "USER_INPUT_CDS_PY_EXEC_POD_IP",
      "git-url": "USER_INPUT_GIT_URL",
      "git-access-token": "USER_INPUT_GIT_ACCESS_TOKEN",
      "git-project-id": "USER_INPUT_GIT_PROJECT_ID",
      "git-branch": "USER_INPUT_GIT_BRANCH",
      "git-download-folder": "/opt/app/onap/blueprints/deploy",
      "git-archive-file-name": "USER_INPUT_GIT_ARCHIVE_FILE_NAME",
      "terraform-var-file-name": "USER_INPUT_TERRAFORM_VARS_FILE_NAME",
      "terraform-plan-folder": "USER_INPUT_TERRAFORM_PLAN_FOLDER",
      "terraform-variable-override": {
        "USER_INPUT_VAR_NAME_1": "USER_INPUT_VAR_VALUE_1",
        "USER_INPUT_VAR_NAME_2": "USER_INPUT_VAR_VALUE_2"
      },
      "terraform-environment-variables": {
        "ARM_SUBSCRIPTION_ID": "USER_INPUT_ARM_SUBSCRIPTION_ID",
        "ARM_CLIENT_ID": "USER_INPUT_ARM_CLIENT_ID",
        "ARM_CLIENT_SECRET": "USER_INPUT_ARM_CLIENT_SECRET",
        "ARM_TENANT_ID": "USER_INPUT_ARM_TENANT_ID",
        "ARM_ACCESS_KEY": "USER_INPUT_ARM_ACCESS_KEY"
      },
      "terraform-action": "apply",
      "terraform-workspace-name": "USER_INPUT_TERRAFORM_WORKSPACE_NAME"
    }
  }
}
```

```
{
  "commonHeader": {
    "originatorId": "System",
    "requestId": "123456",
    "subRequestId": "1234-12234"
  },
  "actionIdentifiers": {
    "blueprintName": "terraform-plan-executor",
    "blueprintVersion": "1.0.0",
    "actionName": "remote-python",
    "mode": "sync"
  },
  "payload": {
    "remote-python-request": {
      "cds-py-exec-pod-ip": "USER_INPUT_CDS_PY_EXEC_POD_IP",
      "git-url": "USER_INPUT_GIT_URL",
      "git-access-token": "USER_INPUT_GIT_ACCESS_TOKEN",
      "git-project-id": "USER_INPUT_GIT_PROJECT_ID",
      "git-branch": "USER_INPUT_GIT_BRANCH",
```

```

        "git-archive-file-name": "USER_INPUT_GIT_ARCHIVE_FILE_NAME",
        "git-download-folder": "USER_INPUT_GIT_DOWNLOAD_FOLDER",
        "terraform-plan-folder": "USER_INPUT_TERRAFORM_PLAN_FOLDER",
        "terraform-var-file-name": "USER_INPUT_TERRAFORM_VARS_FILE_NAME",
        "terraform-environment-variables": {
        },
        "terraform-variable-override": {
            equinix_client_id: "USER_INPUT_EQUINIX_CLIENT_ID",
            equinix_client_secret: "USER_INPUT_EQUINIX_CLIENT_SECRET",
            equinix_port_name: "USER_INPUT_EQUINIX_PORT_NAME",
            aws_access_key: "USER_INPUT_AWS_ACCESS_KEY",
            aws_secret_key: "USER_INPUT_AWS_SECRET_KEY",
            aws_region: "USER_INPUT_AWS_REGION",
            aws_metro_code: "USER_INPUT_AWS_METRO_CODE",
            aws_account_id: "USER_INPUT_AWS_ACCOUNT_ID"
        },
        "terraform-action": "apply",
        "terraform-workspace-name": "USER_INPUT_TERRAFORM_WORKSPACE_NAME"
    }
}

{
    "commonHeader": {
        "originatorId": "System",
        "requestId": "123456",
        "subRequestId": "1234-12234"
    },
    "actionIdentifiers": {
        "blueprintName": "terraform-plan-executor",
        "blueprintVersion": "1.0.0",
        "actionName": "remote-python",
        "mode": "sync"
    },
    "payload": {
        "remote-python-request": {
            cds_py_exec_pod_ip: "USER_INPUT_CDS_PY_EXEC_POD_IP",
            git_url: "USER_INPUT_GIT_URL",
            git_access_token: "USER_INPUT_GIT_ACCESS_TOKEN",
            git_project_id: "USER_INPUT_GIT_PROJECT_ID",
            git_branch: "USER_INPUT_GIT_BRANCH",
            git_archive_file_name: "USER_INPUT_GIT_ARCHIVE_FILE_NAME",
            git_download_folder: "USER_INPUT_GIT_DOWNLOAD_FOLDER",
            terraform_plan_folder: "USER_INPUT_TERRAFORM_PLAN_FOLDER",
            terraform_var_file_name: "USER_INPUT_TERRAFORM_VARS_FILE_NAME",
            "terraform-environment-variables": {
                OS_AUTH_URL: "USER_INPUT_OS_AUTH_URL",
                OS_USERNAME: "USER_INPUT_OS_USERNAME",
                OS_PASSWORD: "USER_INPUT_OS_PASSWORD",
                OS_REGION_NAME: "USER_INPUT_OS_REGION_NAME",
                OS_PROJECT_NAME: "USER_INPUT_OS_PROJECT_NAME",
                OS_PROJECT_DOMAIN_NAME: "USER_INPUT_OS_PROJECT_DOMAIN_NAME",
                OS_USER_DOMAIN_NAME: "USER_INPUT_OS_USER_DOMAIN_NAME",
                OS_IDENTITY_API_VERSION:
"USER_INPUT_OS_IDENTITY_API_VERSION"
            },
            "terraform-variable-override": {
                USER_INPUT_VARIABLE_NAME: "USER_INPUT_VARIABLE_VALUE"
            },
            "terraform-action": "apply",
            "terraform-workspace-name": "USER_INPUT_TERRAFORM_WORKSPACE_NAME"
        }
    }
}

```

Helm Chart Onboarding for Service and App Registration

```

# You can refer to the below EMCO REST API link to get more details about the JSON payload
# https://wiki.onap.org/display/DW/V2+API+Specification
# Note:- You should update the attribute values for USER_INPUT_XXX parameters
# USER_INPUT_AMCOP_APP_API_URL : here update the amcop installed machine ip with portno(30480).
# ex : http://<amcop server ip>:30480/middleend
# USER_INPUT_CDS_PY_EXEC_POD_IP : Use below command to get the CDS pyexecutor IP
# kubectl get pods -n onap -o wide | grep dev-cds-py-executor | awk '{print $6}'
# USER_INPUT_GIT_URL - helm chart git repo url ex. "https://gitlab.com/api/v4"
# USER_INPUT_GIT_PROJECT_ID - project id ex."26902714"
# USER_INPUT_GIT_BRANCH - git branch name ex. "main"
# USER_INPUT_GIT_ARCHIVE_FILE_NAME - any tar file name ex. "helm-charts-poc.tar.gz"
# USER_INPUT_GIT_ACCESS_TOKEN - refer main readme to generate the git access token.
# USER_INPUT_PROJECT_NAME - Enter the existing amcop project name.
# USER_INPUT_SERVICE_NAME - Enter the service name ex:"PCEI-AZURE-IOT-SVC"

$ vi request-payload.json
{
  "commonHeader": {
    "originatorId": "System",
    "requestId": "12345678",
    "subRequestId": "1234-122346"
  },
  "actionIdentifiers": {
    "mode": "sync",
    "blueprintName": "helm-chart-processor",
    "blueprintVersion": "9.0.1",
    "actionName": "helm-chart-prepare"
  },
  "payload": {
    "helm-chart-prepare-request": {
      "cds-py-exec-pod-ip": "USER_INPUT_CDS_PY_EXEC_POD_IP",
      "git-url": "USER_INPUT_GIT_URL",
      "git-access-token": "USER_INPUT_GIT_ACCESS_TOKEN",
      "git-project-id": "USER_INPUT_GIT_PROJECT_ID",
      "git-branch-name": "USER_INPUT_GIT_BRANCH",
      "git-download-folder": "/opt/app/onap/python/test001",
      "git-archive-file-name": "USER_INPUT_GIT_ARCHIVE_FILE_NAME",
      "amcop-middle-end-properties": {
        "create-composit-app-api-url": "USER_INPUT_AMCOP_APP_API_URL",
        "amcop-middle-end-username": "",
        "amcop-middle-end-password": ""
      },
      "amcop-service-information": {
        "project-name": "USER_INPUT_PROJECT_NAME",
        "service-name": "USER_INPUT_SERVICE_NAME",
        "service-description": "USER_INPUT_SERVICE_DESCRIPTION"
      },
      "helm-charts-info": [{
        "helm-chart-folder": "USER_INPUT_HELM_CHART_FOLDER",
        "helm-chart-file-name": "USER_INPUT_HELM_CHART_FILE_NAME",
        "helm-values-yaml-override-properties": {
          "resources.saml1.limits.cpu": "USER_INPUT_RESOURCES_SMALL_LIMITES_CPU",
          "service.type": "USER_INPUT_SERVICE_TYPE",
          "service.internalPort": "USER_INPUT_SERVICE_INTERNALPORT"
        }
      }]
    }
  }
}

```

Service Instance creation and App deployment onto target Kubernetes clusters API

```

{
  "actionIdentifiers": {
    "mode": "sync",
    "blueprintName": "composite-app-deploy-processor",
    "blueprintVersion": "1.0.0",
    "actionName": "simple-composite-app-deploy-workflow"
  },
  "commonHeader": {
    "subRequestId": "143748f9-3cd5-4910-81c9-a4601ff2dc58",
    "requestId": "e5eb1f1e-3386-435d-b290-d49d8af8da4c",
    "originatorId": "SDNC_DG"
  },
  "payload": {
    "simple-composite-app-deploy-workflow-request": {
      "composite-app-deploy-input-params": {
        "composite-application-parameters": {
          "service-instance-name": "USER_INPUT_TARGET_CLUSTER_NAME",
          "service-instance-description": "USER_INPUT_TARGET_CLUSTER_DESCRIPTION",
          "service-instance-version": "USER_INPUT_TARGET_CLUSTER_VERSION",
          "composite-app": "USER_INPUT_COMPOSITE_APP_NAME",
          "composite-app-version": "USER_INPUT_COMPOSITE_APP_VERSION",
          "composite-profile": "USER_INPUT_COMPOSITE_PROFILE"
        },
        "application-data-parameters": [{
          "app-metadata-parameters": {
            "app-name": "USER_INPUT_APP_NAME"
          },
          "target-cluster-provider-parameters": [{
            "cluster-provider-name": "USER_INPUT_PROVIDER_NAME",
            "clusters": [{
              "name": "USER_INPUT_CLUSTER_NAME"
            }]
          }]
        }],
        "application-override-parameters": [{
          "app-name": "USER_INPUT_APP_NAME",
          "values": {
            "USER_INPUT_VARIABLE_NAME": "USER_INPUT_VARIABLE_VALUE"
          }
        }],
        "amcop-middle-end-properties": {
          "deploy-intent-group-api-url": "USER_INPUT_AMCOP_APP_API_URL",
          "amcop-middle-end-username": "",
          "amcop-middle-end-password": ""
        },
        "amcop-service-project-name": "USER_INPUT_AMCOP_SERVICE_PROJECT_NAME"
      }
    }
  }
}

```

PCEI R5 API Postman Collection

<https://gitlab.com/akraino-pcei-onap-cds/equinix-pcei-poc/-/tree/main/postman-collections>