New Hire Ramp Up

Welcome to join our team.

Contents

[Day1. Overall Preparations 3](#_Toc449419976)

[1. Team warm up 3](#_Toc449419977)

[2. Setup machine 3](#_Toc449419978)

[1) After introduction, you need prepare your working environment. 3](#_Toc449419979)

[2) And setup account for our systems: Jira / Git / Jenkins / vCD / Team DL 3](#_Toc449419980)

[3) Install tomcat / notepad++ / sublime / firefox / chrome / safari … 3](#_Toc449419981)

[3. Basic introduction for our daily working. 3](#_Toc449419982)

[1) Production 3](#_Toc449419983)

[2) Process 3](#_Toc449419984)

[4. How to apply leave request 3](#_Toc449419985)

[Day 2. Whole project process and our daily working 4](#_Toc449419986)

[1. Release lifecycle 4](#_Toc449419987)

[1) Overview 4](#_Toc449419988)

[2) Detailed process 4](#_Toc449419989)

[2. Branching strategy 4](#_Toc449419990)

[3. Merge / rebase strategy 4](#_Toc449419991)

[4. Jira 4](#_Toc449419992)

[5. Git 4](#_Toc449419993)

[Day 3 - Week 1. Tasks: DevOps 5](#_Toc449419994)

[1. Tasks 5](#_Toc449419995)

[Week 2. Knowledge: Jenkins build 7](#_Toc449419996)

[1. Build type 7](#_Toc449419997)

[2. Build infrastructure 7](#_Toc449419998)

[3. Create a job 7](#_Toc449419999)

[1) Job template 7](#_Toc449420000)

[2) Info: naming convention 7](#_Toc449420001)

[4. Ref doc 7](#_Toc449420002)

[Week 3. Tasks: Release 8](#_Toc449420003)

[1. Release process 8](#_Toc449420004)

[2. Sub-processes 8](#_Toc449420005)

[3. Ref doc 8](#_Toc449420006)

[Week 4. Knowledge: Git, Gerrit 9](#_Toc449420007)

[1. How to use GIT 9](#_Toc449420008)

[2. Git best practice 9](#_Toc449420009)

[3. Playground info: Git servers list. 9](#_Toc449420010)

[Week 5. Knowledge: Maven, Nexus 10](#_Toc449420011)

[1. Maven 10](#_Toc449420012)

[2. Nexus 10](#_Toc449420013)

[PS. Miscellaneous 11](#_Toc449420014)

# Day1. Overall Preparations

## Team warm up

* Introduce team member
* Nice tradition
* Tour around the office

Tips: you can start from Point two if you buddy is busy.

## Setup machine

### After introduction, you need prepare your working environment.

Such as: Email / printer / name template / Lync

* Lync just like QQ, it is already installed and log on with your account.

### And setup account for our systems: Jira / Git / Jenkins / vCD / Team DL

Short introduce for them:

Jira: Normally we log our task, bug in Jira.

Git: Git is a file version management tool. All the resource like code, doc will be stored in GIT. Dev will check in their code in GIT too.

Jenkins: used for automation work.

vCD: vCloud Director, a virtual machine management page.

### Install tomcat / notepad++ / sublime / firefox / chrome / safari …

Tips: **If you can’t access the share folder which start with (**[**\\...\**](file:///\\...\)**...) call your buddy for help**

## Basic introduction for our daily working.

### Production

### Process

## How to apply leave request

# Day 2. Whole project process and our daily working

First of all, review the content in Day.1

Today we will introduce our release lifecycle based on scrum framework, how to use **Jira**, install **Git** and our major project **etc.**

## Release lifecycle

### Overview

### Detailed process

## Branching strategy

## Merge / rebase strategy

## Jira

Jira is a tool to manage tasks and bugs. Learn how to create/close bugs or tasks.

## Git

Refer to Day 6. Knowledge: Git, Gerrit.

Practice:

1. Understand the processes (listed above).
2. Create a feature branch
3. Close a feature branch

Expected Result:

1. Tasks done.
2. Processes (listed above) are understood.

# Day 3 - Week 1. Tasks: DevOps

## Tasks

Following tasks are among the DevOps jobs while not limited.

|  |  |  |
| --- | --- | --- |
|  | Task | Working Instruction |
| SCM | Create branch  Close branch |  |
| Role (user / group) management |  |
|  |  |  |
| Build | Monitor the build and release infrastructure |  |
| Set up a Jenkins slave |  |
| Set up a Jenkins job |  |
| Back up Jenkins master |  |
| Create a snapshot of a build vm |  |
| Update Jenkins master and build machines |  |
|  |  |  |
| Release | Release a version |  |
| Static code scan |  |
| Fortify scan |  |
| Portal update |  |
| Prepare legal scan |  |
|  | Other quality metric job |  |
|  |  |  |
| Dev Support | Dev support |  |
|  |  |  |

Practice:

In a **test** environment,

1. Practice the SCM tasks in above table.
2. Practice the Build tasks in above table.
3. Practice the Dev Support tasks in above table.

Expected Result:

1. Tasks done.
2. Understand why / how it works.
3. Clean up the test environment.

# Week 2. Knowledge: Jenkins build

## Build type

Table: Jenkins Build type

|  |  |  |
| --- | --- | --- |
| Build Type | Purpose | Notes |
| Continuous Integration  (ex. byChk) |  |  |
| Nightly |  |  |
| Voter |  |  |
| deployPortal / snippet |  |  |
| Quality (jscover, fortify, etc.) |  |  |

## Build infrastructure

## Create a job

### Job template

### Info: naming convention

Branch

Build

## Ref doc

**Practice**:

1. Fill up the Jenkins Build type table based on your understanding.
2. Create wiki for tasks in Create a job.
3. Create an own workspace, play all the steps in a build job locally.

Expected Result:

1. Tasks done.
2. Own development environment is set up well.

# Week 3. Tasks: Release

## Release process

## Sub-processes

## Ref doc

**Practice**:

1. Read through the processes, documents, write a brief summary.
2. Do a release when release request comes.
3. Write an introduction of the build process in …environment.

Expected Result:

1. Tasks done.
2. Process is understood and summary written.
3. introduction is written.

# Week 4. Knowledge: Git, Gerrit

## How to use GIT

1. Clone the repository for you project.
2. Git repository
3. Check out the remote branch
4. GIT commit operation.

## Git best practice

|  |  |
| --- | --- |
| Best Practice | WI |
| Set up a git client |  |
| How to do git commit |  |
| How to download a patch set |  |
| How to merge |  |
| Gerrit Review |  |
| Gerrit Role Management |  |

## Playground info: Git servers list.

|  |  |  |
| --- | --- | --- |
| Git Server | Purpose | Notes |
|  |  |  |
|  |  |  |

Practice:

1. Practice in a playground: check out, commit, update a commit
2. Practice merge, rebase, cherry-pick etc.
3. Practice code review in a playground.

Expected Result:

1. Practice done.
2. Local environment for Git, Gerrit actions (listed above) works.
3. Environment for Git, Gerrit actions (listed above) on a jenkins slave works

# Week 5. Knowledge: Maven, Nexus

## Maven

1. Maven Project Guide
2. How to Update POM Version
3. Info: how to check an effective pom

<http://maven.apache.org/plugins/maven-help-plugin/effective-pom-mojo.html>

## Nexus

**Practice**:

1. Read through the documents, write a brief summary of:

Maven process.

Maven and nexus interaction

1. Do maven build locally in own workspace
2. Check / configure nexus settings ( and / or proxy) in a Jenkins job

Expected Result:

1. Practice done.
2. Maven, Nexus purpose, daily usage is understood and summary is written.
3. Maven build locally in own workspace works.
4. Maven build on a jenkins slave works.

# PS. Miscellaneous

Etc.