DEVELOPER

PLAY BOOK - ROAD TO WIKI

WHAT IS BASIC PROGRAMMING CONCEPTS?

1. Variables, Data Types, and Operators:

Variables:

- Named containers that store data.
- Think of them as labelled boxes.

Data Types:

- Define the kind of data a variable can hold (numbers, text, true/false).

Operators:

- Symbols that perform actions on data.
- Examples: maths (+, -, *, /), comparisons (==, !=, <, >), logic (and, or, not).

2. Control Structures: Conditionals and Loops:

Conditionals (if/else statements):

- Make decisions based on conditions (true/false).
    - If true, execute one code block.
    - If false, execute another.

Loops (for/while loops):

- Repeat a block of code multiple times.

For loops:

Know beforehand how many times to repeat.

While loops:

Repeat until a condition is met.

3. Functions and Scope:

- **Functions:**
    - Reusable code blocks that perform tasks.

    - Take inputs (parameters) and can return outputs.
    - Promote code organisation and reusability.
- **Scope:**
    - Accessibility of variables and functions.
    - Local variables: accessible only within their defining function.
    - Global variables: accessible from anywhere (use cautiously!).

4. Basic Data Structures: Arrays, Lists, and Dictionaries:

- **Arrays:**
    - Ordered collections of items (all the same data type).
    - Imagine an array as a fixed-size shelf with a specific order.
- **Lists:**
    - Similar to arrays, but flexible in size and data type.
    - Think of a list as a backpack where you can add various items.
- **Dictionaries:**
    - Unordered collections of key-value pairs.
    - Function like phonebooks: find a value (phone number) using a key (name).

You can find the above informations in detail here:- click here

WHAT IS WEB DEVELOPMENT BASICS?

1. HTML (HyperText Markup Language)

- **Structure of HTML Documents:** Understanding elements, tags, attributes, and nesting.
- **Common HTML Elements:** `<header>`, `<footer>`, `<article>`, `<section>`, `<div>`, `<span>`, `<form>`, `<input>`, `<button>`, etc.
- **Forms and Inputs:** Creating forms, handling user input, and form validation.
- **Semantic HTML:** Using semantic elements to improve accessibility and SEO.

Resources:

- HTML5 Basics - Introduction to HTML5.

2. CSS (Cascading Style Sheets)

- **CSS Syntax:** Selectors, properties, and values.
- **Box Model:** Margin, border, padding, and content.
- **Positioning:** Static, relative, absolute, and fixed positioning.

- **Flexbox and Grid Layouts:** Creating responsive and flexible layouts.
- **Responsive Design:** Media queries, mobile-first design principles.

Resources:

- CSS Basics - Introduction to CSS and styling.

3.JavaScript and DOM Manipulation

- JavaScript functions and events.
- Manipulating the DOM.
- Simple interactive web page creation.

Resources:

- JavaScript Basics: Learn the fundamentals of JavaScript and how to manipulate the DOM.

4. HTTP and Web Browsers

- **Understanding HTTP:** Methods (GET, POST, PUT, DELETE), status codes, headers.
- **Request/Response Cycle:** How browsers communicate with web servers.
- **Cookies and Sessions:** Basic concepts of cookies and sessions for maintaining user state.

Resources:

- HTTP Overview - Introduction to HTTP protocol.

5. Development Tools

- **Text Editors and IDEs:** Choosing and configuring text editors like VS Code, Sublime Text.
- **Browser DevTools:** Using developer tools in browsers for debugging and performance analysis.
- **Package Managers:** Introduction to npm/yarn for managing dependencies.

Resources:

- VS Code Documentation - Guide to using Visual Studio Code.

WHAT IS VERSION CONTROL?

- **Introduction to Git:** Setting up Git, basic commands (init, add, commit, push, pull).
- **Branching and Merging:** Creating branches, merging changes, resolving conflicts.
- **Using GitHub:** Pushing code to GitHub, collaborating on projects.

Resources:

- Git Basics - Comprehensive guide to using Git.
- Git Basics[video] - Basic understanding of Git: repositories, commits, branches, and merging.

WHAT IS DATABASE BASIC?

- **Understanding Relational Databases**
- **Basic SQL Queries:** SELECT, INSERT, UPDATE, DELETE.

Resources:

- Database Basics: An overview of relational databases and fundamental SQL operations.
- SQL Tutorial: A comprehensive guide to SQL with examples and exercises.
- Introduction to Databases by Khan Academy: Free course covering SQL basics and database management.
- MySQL for Beginners: MySQL resources for beginners.

INTRODUCTION TO COMMAND LINE INTERFACE

- Navigating the file system using CLI.
- Basic file and directory operations.
- Practical Git session: Cloning a repository, making changes, committing, and pushing.

Resources:

- CLI Basics - Navigation
- CLI Basics - File Management
- Resource: CLI Basics - File Viewing
- Resource: Introduction to Git - CLI Git Commands

- Resource: Introduction to Bash Scripting
- Command Line Interface Basis [Video]

WHAT ARE THE BASICS OF WIKI? (Compulsory)

Essential Training Resources (Very Important)

- Wikimedia Technology Training 2024: Developers Ecosystem: Video resources covering the Wikimedia Developer Ecosystem.
- Wikimedia Technology Training 2024: MediaWiki Databases: Training videos on MediaWiki databases.
- Wikimedia Technology Training 2024: User Scripts: Learn about user scripts in MediaWiki through these videos.
- Wikimedia Technology Training 2024: APIs: Overview of Wikimedia APIs for developers.
- Introduction to Wikimedia Cloud Services: Video introduction to Wikimedia Cloud Services.
- Wikifunctions : Overview of Wikifunctions, a project to create and share functions as a new type of content
- First steps with Wikifunctions: Wikimedia Technology Training 2024: First Steps with Wikifunctions.

HOW TO SET UP MEDIA WIKI

Step 1: Download MediaWiki

Download the latest stable release of MediaWiki from the official website:

- MediaWiki Downloads

Step 2: Extract Files

- Extract the downloaded MediaWiki files to your web server's root directory.

Step 3: Configure Database

Set up your database and user:

- Create a new database for MediaWiki.
- Create a user and grant necessary permissions to the database.

Step 4: Run the Installer

Access the MediaWiki installer by navigating to your server's URL:

- Open your web browser and go to `http://yourserverIP/MediaWiki`
- Follow the on-screen instructions to configure your wiki.

Step 5: Configure LocalSettings.php

After completing the installation, the installer will generate a `LocalSettings.php` file. Download this file and place it in the root directory of your MediaWiki installation.

Step 6: Final Configuration

Fine-tune your MediaWiki configuration by editing the `LocalSettings.php` file:

- Set up email and user authentication.
- Configure caching for better performance.
- Customise the appearance and functionality with extensions and skins.

Step 7: Install Extensions and Skins

Enhance your MediaWiki installation by adding extensions and skins:

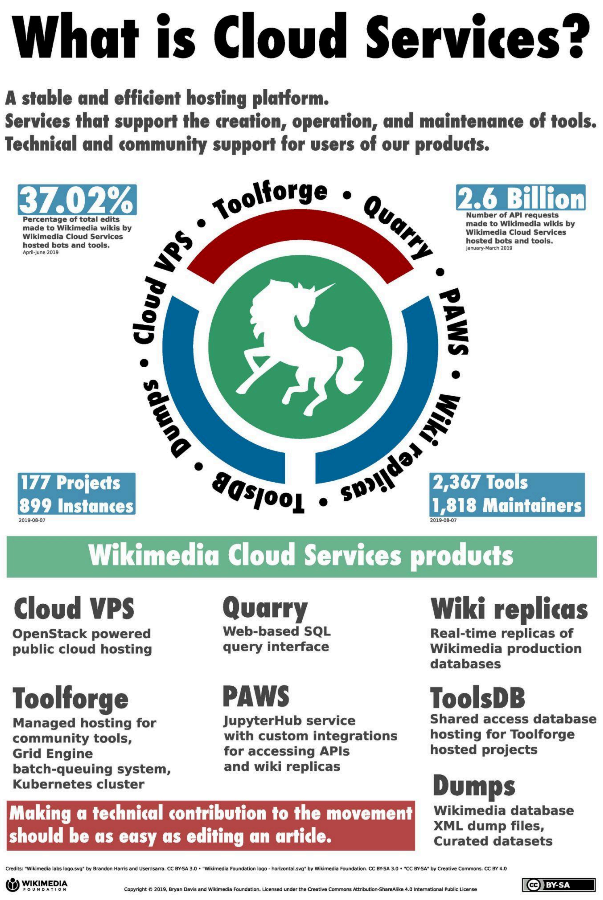
- MediaWiki Extensions
- MediaWiki Skins

Resources:

- Introduction to MediaWiki : Core concepts (Part 1): Overview of fundamental MediaWiki concepts and how they work.
- Introduction to MediaWiki : Wikipedia's extensions (Part 2): Insight into popular extensions used by Wikipedia and their functionalities.
- How to set up mediawiki: Step-by-step guide to installing and configuring MediaWiki.
- Setup MediaWiki - Video Guide: A comprehensive video tutorial on setting up MediaWiki.
- Setup MediaWiki - Detailed Video Guide: An in-depth video covering additional setup details.

- Phabricator Help: Guide to using Phabricator for project management and issue tracking.
- Developer Hub: Central hub for resources and documentation for developers.
- Support Channels & Youtube : Channels for getting help and support within the Wikimedia community.
- New developer: Resources and guidance for new developers entering the Wikimedia ecosystem.

WIKIMEDIA CLOUD SERVICES



What is Cloud Services?

A stable and efficient hosting platform.
Services that support the creation, operation, and maintenance of tools.
Technical and community support for users of our products.

37.02% Percentage of total edits made to Wikimedia wikis by Wikimedia Cloud Services hosted bots and tools. April-June 2019

2.6 Billion Number of API requests made to Wikimedia wikis by Wikimedia Cloud Services hosted bots and tools. January-March 2019

177 Projects **899 Instances** 2019-08-07

2,367 Tools **1,818 Maintainers** 2019-08-07

Wikimedia Cloud Services products

Cloud VPS OpenStack powered public cloud hosting

Quarry Web-based SQL query interface

Wiki replicas Real-time replicas of Wikimedia production databases

Toolforge Managed hosting for community tools, Grid Engine batch-queuing system, Kubernetes cluster

PAWS JupyterHub service with custom integrations for accessing APIs and wiki replicas

ToolsDB Shared access database hosting for Toolforge hosted projects

Dumps Wikimedia database XML dump files, Curated datasets

Making a technical contribution to the movement should be as easy as editing an article.

Wikimedia Cloud Services (WCS) is a suite of tools, services, and infrastructure designed to support technical contributors in building and maintaining tools, bots, web services, and other projects that improve Wikimedia projects.

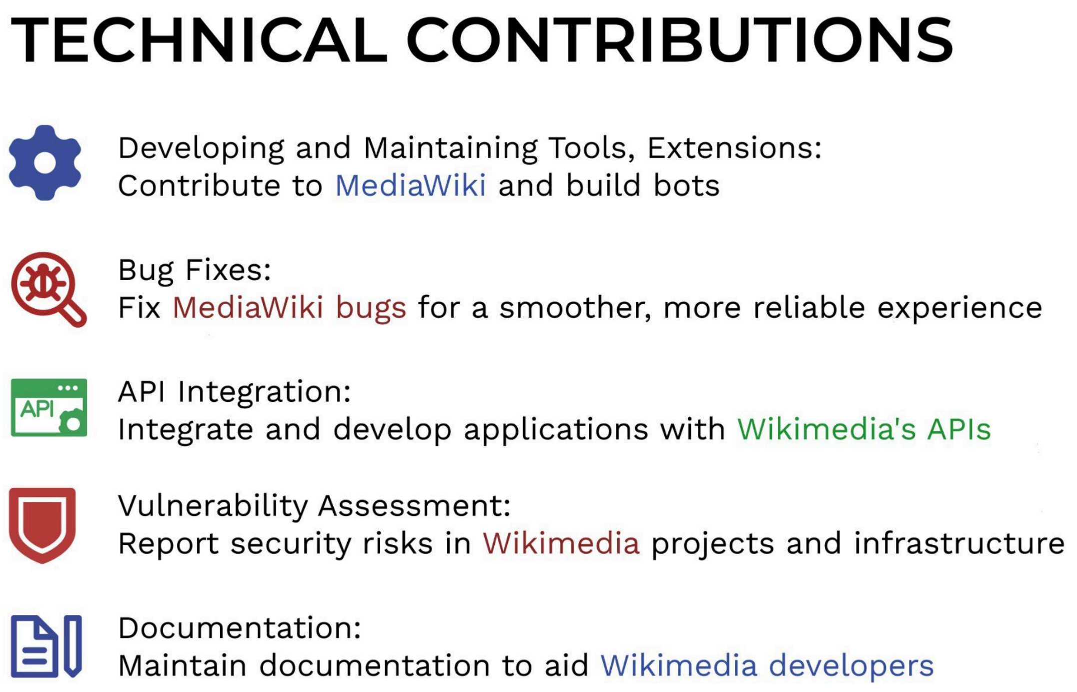
Resources:

- Wikidata Query Service : Tool for querying and extracting data from Wikidata.

- Help:Cloud Services introduction : Overview and guidance for using Wikimedia Cloud Services.
- Wikimedia Cloud VPS Portal: Overview and Resources: The Wikimedia Cloud VPS portal provides information and resources for managing and using cloud-based virtual private servers within the Wikimedia ecosystem.
- Where can I run this? An introduction to Wikimedia Cloud Services: Wikimedia Cloud Services provides a comprehensive platform for running and managing cloud-based tools and applications within the Wikimedia ecosystem.
- Commons:Structured data : Information on structured data implementation and usage on Wikimedia Commons.

Additional resources:

- Wikimedia Tech Safari Program/Documentation : Documentation for the Wikimedia Tech Safari program.



TECHNICAL CONTRIBUTIONS

Developing and Maintaining Tools, Extensions:
Contribute to MediaWiki and build bots

Bug Fixes:
Fix MediaWiki bugs for a smoother, more reliable experience

API Integration:
Integrate and develop applications with Wikimedia's APIs

Vulnerability Assessment:
Report security risks in Wikimedia projects and infrastructure

Documentation:
Maintain documentation to aid Wikimedia developers

HOW TO CREATE WIKIMEDIA DEVELOPER ACCOUNT?

1. **Visit the Wikimedia Developer Account Creation Page**
    - Go to the Create a Wikimedia Developer Account page. This page provides detailed instructions on how to set up your account.
2. **Understanding Phabricator**
    - Phabricator is the project management tool used by Wikimedia developers. Familiarise yourself with Phabricator by visiting the Phabricator Help page. This guide will help you understand how to use Phabricator effectively.
    - Phabricator Account Setup: Make your Phabricator Account
3. **Setting Up Gerrit**
    - Gerrit is the code review tool used by Wikimedia. You will need to set up Gerrit as part of your developer account creation process. Visit the Gerrit page for instructions on how to configure Gerrit.
4. **Familiarise Yourself with the Developer Hub**
    - The Developer Hub is a central resource for Wikimedia developers. It contains all the necessary documentation and resources you will need as a developer.
    - How to become a MediaWiki hacker : Tips and resources for becoming proficient in MediaWiki development and ecosystem behind wikimedia
5. **Explore Good First Bugs**
    - To get started with your first contribution, check out the Good First Bugs page. This list contains beginner-friendly bugs that are ideal for new contributors.
6. **Read the How to Contribute Guide**
    - Visit the How to Contribute page. This guide provides an overview of the various ways you can contribute to Wikimedia projects.
7. **Join the New Developers Program**
    - If you are new to Wikimedia development, consider joining the New Developers program. This program provides mentorship and resources to help new developers get started.
    - Hackathons/Handbook/Newcomers : Comprehensive guide for newcomers participating in Wikimedia hackathons

HOW DO YOU MAKE AN ACCOUNT ON GERRIT?

Creating an account on Gerrit is essential for contributing code and participating in code reviews within the Wikimedia ecosystem. Follow these steps to set up your Gerrit account:

1. Create a Wikimedia Developer Account

- **Description:** Before creating a Gerrit account, you need a Wikimedia developer account. Ensure you have registered and set up your developer account.
- **Link:** Create a Wikimedia Developer Account - Wikitech

2. Access Gerrit Registration

- **Description:** Navigate to the Gerrit settings page to start the account creation process.
- **To create a Gerrit account for code review and version control management and add a Phabricator ID:** Make your Gerrit Account

---

Important instructions
Command Prompt and Powershell are NOT recommended. Instead use **Git Bash Terminal** or WSL to run commands.

---

3. Add Your SSH Key to Gerrit

- **Description:** You need to add your SSH key to Gerrit to authenticate and interact with the repositories. Follow the instructions to upload your SSH key.
- **Link:** Add your SSH Key here / Add SSH Key in Gerrit [Guide]

4. Set Up SSH Keys

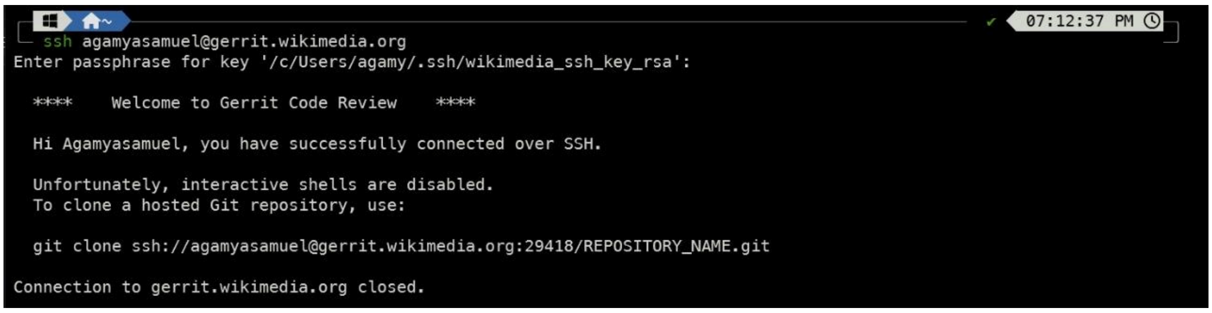
- **Description:** For secure communication with Gerrit, set up SSH keys. Although this is not part of the Gerrit account creation, it's crucial for contributing code.
- **Video Demo:** How to Generate and Add SSH Key to Your GitLab Account
- **Part 1 - Setup SSH for GitHub Account:** Setup SSH for GitHub
- **Part 2 - Config SSH with Custom SSH Key Git and GitHub:** Config SSH with Custom SSH Key
- **Configuring .ssh Config File:** Detailed setup for multiple Git providers.
    - **Location on Windows:**
      `C:/users/your_laptop_user_name/.ssh/config`

Example Configuration:

```
Host gerrit.wikimedia.org
        HostName gerrit.wikimedia.org
        Port 29418
        User hridyeshgupta
        IdentityFile ~/.ssh/id_rsa
        IdentitiesOnly yes
```

5. Test Your SSH Key

- **Description:** Verify that your SSH key is correctly set up and can connect to Gerrit.
- **Command:**
  `ssh -p 29418 your_gerrit_username@gerrit.wikimedia.org`
- **Expected Response:** Successful connection indicates proper configuration.



```
 ssh agamyasamuel@gerrit.wikimedia.org                                07:12:37 PM
Enter passphrase for key '/c/Users/agamy/.ssh/wikimedia_ssh_key_rsa':

  ****    Welcome to Gerrit Code Review    ****

  Hi Agamyasamuel, you have successfully connected over SSH.

  Unfortunately, interactive shells are disabled.
  To clone a hosted Git repository, use:

  git clone ssh://agamyasamuel@gerrit.wikimedia.org:29418/REPOSITORY_NAME.git

Connection to gerrit.wikimedia.org closed.
```

`If you get this Response, then your SSH is configured properly.`

6. Familiarise Yourself with Gerrit

- **Description:** Understand Gerrit's interface and functionalities to effectively participate in code reviews.
- **Tutorial:** Gerrit/Tutorial - MediaWiki

7. Learn More About Gerrit

- **Description:** Explore in-depth resources and courses on Gerrit to enhance your Git skills.
- **Course:** Using Gerrit to Enhance Your Git
- Differences between Gerrit and standard Git commands.

HOW TO ADD SSH KEY TO GERRIT?

Adding an SSH key to Gerrit is essential for secure authentication and to push code changes. Follow these steps to set up and add your SSH key:

1. Generate an SSH Key Pair (if you haven't already):

- **Description:** An SSH key pair is needed to securely connect to Gerrit. If you don't have an SSH key pair, generate one using your terminal.

Resource:

- How to Generate and Add SSH Key to Your GitLab Account *(Video demo for GitLab, but steps are similar for Gerrit)*
- *Generate GitLab SSH Keys & setup SSH on GitLab (Video demo for GitLab, but steps are similar for Gerrit)*

2. Copy Your SSH Public Key:

- **Description:** Once you've generated your SSH key pair, copy the contents of the public key file (usually `id_rsa.pub`).
- **Command:** `cat ~/.ssh/id_rsa.pub` *(This will display your public key in the terminal)*

3. Access Gerrit's SSH Key Settings:

- **Description:** Login to Gerrit and navigate to the settings page where you can add your SSH key.

4. Add Your SSH Key to Gerrit:

- **Description:** On the Gerrit SSH keys settings page, paste your copied SSH public key into the designated field and save.

5. Test Your SSH Key Configuration:

- **Description:** Ensure your SSH key is properly configured by testing the connection to Gerrit.
- **Command:**
  ```
  ssh -p 29418 your_gerrit_username@gerrit.wikimedia.org
  ```
- **Expected Result:** If configured correctly, you should receive a successful response.

6.Add your ssh key here in Gerrit:

- Instructions for adding your SSH key to Gerrit at
- https://gerrit.wikimedia.org/r/settings/#SSHKeys

HOW TO PUSH CHANGES ON GERRIT?

Pushing changes to Gerrit involves a few key steps to ensure your code is reviewed and merged properly. Follow these instructions to push your changes effectively.

1. Prepare Your Local Repository

Description: Ensure your local Git repository is up-to-date and that you have committed all necessary changes before pushing to Gerrit.

Steps:

- Open your Git terminal (Git Bash or WSL).
- Navigate to your local repository using the `cd` command.

Resources:

- Git Basics - A Command Line Tutorial

2. Add Your Changes

Description: Stage the changes you want to push to Gerrit.

Command:

```
git add .
```

This command stages all changes in the repository. You can also specify individual files if needed.

Resources:

- Git Add Documentation

3. Commit Your Changes

Description: Commit your changes with a descriptive message to explain what was fixed or added.

Command:

```
git commit -m "Describe what you fixed in your commit"
```

Replace the placeholder text with a meaningful commit message.

Resources:

- Git Commit Documentation
- Gerrit Commit Message Guidelines

4. Push Your Changes to Gerrit

Description: Push your committed changes to Gerrit for review.

Command:

```
git push origin HEAD:refs/for/master
```

Use `refs/for/master` if `master` is the primary branch. If your project uses a different primary branch, replace `master` with the appropriate branch name, such as `main`.

Additional Command for Branches with Different Names:

```
git push origin HEAD:refs/for/main
```

Resources:

- Pushing changes to Gerrit

5. Verify Your Changes

Description: After pushing, verify that your changes appear in Gerrit and are ready for review.

Steps:

- Visit Gerrit Code Review to check your changes.
- Ensure that your changes are correctly listed and that no errors occurred during the push.

Resources:

- Gerrit/Tutorial - MediaWiki

HOW TO TEST YOUR SSH KEY?

Testing your SSH key is crucial to ensure that it is properly configured and can connect to Gerrit. Follow these steps to verify your SSH key setup:

1. Verify SSH Key Configuration:

- **Purpose:** Ensure that your SSH key is correctly set up and can establish a connection with Gerrit.
- **Steps:**

1. Open your terminal or Git Bash.

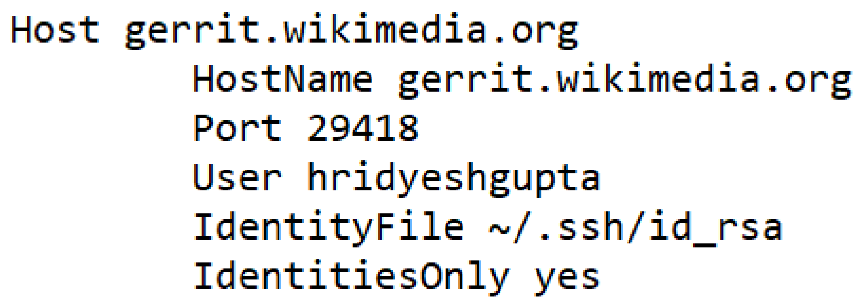
Run the following command to test the SSH connection:
```
ssh -p 29418 your_gerrit_username@gerrit.wikimedia.org
```

2. If the configuration is correct, you should see a message indicating successful authentication.

2. Configure Your SSH Key:

- **Purpose:** Properly set up your `.ssh/config` file for Gerrit and other Git providers.
- **Instructions:**
  - **File Location on Windows:** `C:/users/your_laptop_user_name/.ssh/config`
  - **File Name:** `config` (without extension)

Example Configuration:



```
Host gerrit.wikimedia.org
        HostName gerrit.wikimedia.org
        Port 29418
        User hridyeshgupta
        IdentityFile ~/.ssh/id_rsa
        IdentitiesOnly yes
```

3. Resources:

- **Setting Up .ssh Config File for Gerrit**: Video tutorial for setting up the SSH configuration.(Part1)
- **Setting Up .ssh Config File for Gerrit**: Video tutorial for setting up the SSH configuration.(Part2)
- **How to Generate and Add SSH Key to Your GitLab Account**: Video demo, useful for understanding the process, though it's for GitLab, the steps are similar for Gerrit.
- **SSH Key Testing Command:** Detailed steps and additional information on using Gerrit.

4. Troubleshooting:

- **If you encounter issues:**
  - Ensure that your SSH key is correctly added to Gerrit.
  - Verify that the `.ssh/config` file is properly set up.

- ○ Check for any error messages and consult the relevant documentation or forums for troubleshooting advice.

HOW TO CONTRIBUTE IN AN OPEN SOURCE?

Contributing to open source projects is a valuable way to collaborate with the community, improve your skills, and give back to the software ecosystem. Here's a comprehensive guide on how to get started with contributing to open source, with relevant links and resources:

1. **Understand the Basics of Open Source:**

- **Definition:** Open source software is developed collaboratively and is freely available for anyone to use, modify, and distribute.
- **Learn More:** What is Open Source?

2. **Find a Project to Contribute To:**

- **Explore Wikimedia Projects:**
  - **Good First Bugs - Wikimedia:** A list of beginner-friendly issues to get started with Wikimedia projects.
  - **How to Find Repositories for Your Task :** Guide to finding the right repositories for your contributions.
  - How to contribute **: General guidelines on contributing to open-source projects or specific platforms.**
  - Contributing to Wikimedia software projects **: Guide for developers on how to contribute to Wikimedia's software.**
- **Other Open Source Platforms:**
  - **GitHub Explore:** Discover popular open source projects and issues.
  - **GitLab Explore:** Explore open source projects and contribute.
- Contribute by topic **: Ways to contribute to Wikimedia projects based on specific topics.**
- Outreach programs**:Information on Wikimedia's outreach programs aimed at engaging and supporting new contributors.**

3. **Set Up Your Development Environment:**

- **Basic Tools:**
  - **Git Installation and Setup:** Official Git documentation for installing and configuring Git.

    - **Setting Up SSH Keys:** Video tutorial for generating and adding SSH keys, which is crucial for accessing repositories.
    - **Generate GitLab SSH Keys & setup SSH on GitLab**
- **Preferred Terminal:** Use **Git Bash** or **WSL** for a better experience with Git and Gerrit. Why Git Bash is Preferred

4. Make Your First Contribution:

- **Fork and Clone the Repository:**
    - **How to Fork a Repository**: GitHub guide for forking repositories.
    - **Cloning a Repository**: Git documentation for cloning repositories.
- **Create a New Branch:**
    - **Branching in Git**: Learn about creating and managing branches.
- **Make Changes and Commit:**
    - **Making Commits**: Documentation for committing changes.
- **Push Changes and Create a Pull Request (PR):**
    - **Creating a Pull Request**: GitHub guide for creating pull requests.

5. Review and Address Feedback:

- **Gerrit Code Review**: Overview of code review processes in Gerrit.

6. Engage with the Community:

- **Wikimedia Tech Talk**: Blog and updates on Wikimedia projects.
- **Open Source Forums:** Participate in discussions and get support.

`Commands to Push code (for Gerrit ONLY) -`

```
1)git add .
```

```
2)git commit -m "Describe what you fixed in your commit"
-m “Bug: Phabricator_Ticket_number”
```

- In some cases the Phabricator issue will be a feature or task, in that case don’t write bugs for it instead write Feature/Task.

```
3)git push origin HEAD:refs/for/master
```

- Some branch don’t have “master” branch as their primary branch, some have other name like “main” as their primary branch,
In that case ->

```
git push origin HEAD:refs/for/main
```

What are the Communication Channels?

IN Alias (wikitech@indicwiki.org) You will receive all the regional announcements and communications via this channel.	IN Telegram Channel (https://t.me/+_4ag-VxpVmUxMWI1) To communicate with WIKI Tech from India and hear updates from the regional team.
There will be a regular connect with the Regional Program Team. Details will be shared in your inbox accordingly.	